

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of

Shinichiro TANIGUCHI et al.

Application No.: 09/454,865

Filed: December 7, 1999

For: DISTRIBUTION INFORMATION MANAGEMENT SYSTEM AND METHOD



Group Art Unit: 2734

Docket No.: 104934

Group 2700

FEB 17 2000

RECEIVED

#4
Priority
Page
4-4-00

CLAIM FOR PRIORITY

Assistant Commissioner for Patents
Washington, D.C. 20231.

Sir:

The benefit of the filing date of the following prior foreign application filed in the following foreign country is hereby requested for the above-identified patent application and the priority provided in 35 U.S.C. §119 is hereby claimed:

Japanese Patent Application No. 11-115551 filed April 22, 1999

In support of this claim, a certified copy of said original foreign application:

 X is filed herewith.

 was filed on in Parent Application No. filed .

It is requested that the file of this application be marked to indicate that the requirements of 35 U.S.C. §119 have been fulfilled and that the Patent and Trademark Office kindly acknowledge receipt of this document.

Respectfully submitted,

James A. Oliff
Registration No. 27,075

Thomas J. Pardini
Registration No. 30,411

JAO:TJP/kmc

OLIFF & BERRIDGE, PLC
P.O. Box 19928
Alexandria, Virginia 22320
Telephone: (703) 836-6400

**DEPOSIT ACCOUNT USE
AUTHORIZATION**
Please grant any extension
necessary for entry;
Charge any fee due to our
Deposit Account No. 15-0461

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

RECEIVED
FEB 17 2001
Group 2700

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application:

1999年 4月22日

出 願 番 号
Application Number:

平成11年特許願第115551号

出 願 人
Applicant (s):

富士ゼロックス株式会社

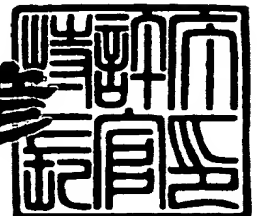


RECEIVED
FEB 17 2000
Group 2700

1999年10月29日

特許庁長官
Commissioner,
Patent Office

近藤 隆彦



出証番号 出証特平11-3074825

【書類名】 特許願

【整理番号】 FN99-00050

【提出日】 平成11年 4月22日

【あて先】 特許庁長官殿

【国際特許分類】 G09G 3/00

【発明の名称】 物流情報管理装置および方法

【請求項の数】 27

【発明者】

 【住所又は居所】 神奈川県足柄上郡中井町境 4 3 0 グリーンテクなかい
 富士ゼロックス株式会社内

 【氏名】 谷口 慎一郎

【発明者】

 【住所又は居所】 神奈川県足柄上郡中井町境 4 3 0 グリーンテクなかい
 富士ゼロックス株式会社内

 【氏名】 河野 健二

【発明者】

 【住所又は居所】 神奈川県足柄上郡中井町境 4 3 0 グリーンテクなかい
 富士ゼロックス株式会社内

 【氏名】 申 吉浩

【特許出願人】

 【識別番号】 000005496

 【氏名又は名称】 富士ゼロックス株式会社

 【電話番号】 0462-38-8516

【代理人】

 【識別番号】 100086531

 【弁理士】

 【氏名又は名称】 澤田 俊夫

 【電話番号】 03-5541-7577

【手数料の表示】

【予納台帳番号】 038818

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 流通情報管理装置および方法

【特許請求の範囲】

【請求項 1】 物品に添付され、上記物品に関する情報を記憶するデータキャリアと、上記データキャリアに情報を読み書きするための物流情報処理モジュールと、上記物品の流通に関する情報を管理する物流情報管理モジュールとを含んで構成されている物流情報管理装置において、

上記物流情報処理モジュールは、

上記データキャリアのデータを読み出す読み取り手段と、

上記データキャリアに情報を書込む書き込み手段と、

上記データキャリアから読み出した情報を検証する第一の情報検証部と、

上記データキャリアへ書込む情報を処理する情報生成部と、

上記物流情報管理モジュールとの通信を行うための第一の通信手段と、

から構成され、

上記第一の情報検証部は、

上記データキャリアから読み出した情報の検証を行う第一の情報検証手段と

上記第一の情報検証手段が情報の検証に用いる検証鍵を記憶する第一の検証鍵記憶手段と、

から構成され、

上記情報生成部は、

上記データキャリアに書込む情報を生成する物流情報生成手段と、

署名生成処理を行う署名モジュールと、

上記署名モジュールが電子署名生成時に使用する署名鍵情報を記憶する署名鍵記憶手段と、

上記署名鍵記憶手段に記憶された署名鍵情報を選択するための署名鍵情報選択手段と、

上記署名鍵情報を物流情報管理モジュールから取得するための署名鍵情報取得手段と、

から構成され、

上記署名モジュールは、

上記物流情報生成手段によって生成された情報に対する電子署名を生成する署名手段と、

上記署名手段が電子署名生成時に使用する署名者秘密情報を記憶する第一の署名者秘密情報記憶手段と、

から構成され、

上記物流情報管理モジュールは、

上記物流情報処理モジュールとの通信を行うための第二の通信手段と、

上記物流情報処理モジュールから送られてきた情報を処理するための第二の情報検証部と、

上記物流情報処理モジュールへ送る署名鍵情報を処理するための署名鍵情報生成部と、

から構成され、

上記第二の情報検証部は、

上記物流情報処理モジュールから送られてきた情報を検証するための第二の情報検証手段と、

上記第二の情報検証手段が情報の検証に用いる検証鍵を記憶する第二の検証鍵記憶手段と、

から構成され、

上記署名鍵情報生成部は、

上記物流情報処理モジュールが物流情報生成時に用いる署名鍵情報を生成する署名鍵情報生成手段と、

上記署名鍵情報生成手段が署名鍵情報生成時に用いる署名鍵を記憶する署名鍵記憶手段と、

上記署名鍵情報生成手段が署名鍵情報生成時に用いる署名者秘密情報を選択するための署名者秘密情報選択手段と、

上記署名者秘密情報を記憶する第二の署名者秘密情報記憶手段と、

から構成される

ことを特徴とする物流情報管理装置。

【請求項 2】 上記署名モジュールは、上記物流情報処理モジュールから取り外し交換することができることを特徴とする請求項 1 の物流情報管理装置。

【請求項 3】 上記署名モジュールは、耐タンパー性を持つことを特徴とする請求項 1 または 2 に記載の物流情報管理装置。

【請求項 4】 上記情報生成部は、署名鍵使用限度情報保持手段を持ち、上記署名鍵情報選択手段は、特定回数以上署名に用いた署名鍵情報を選択しないことを特徴とする請求項 1、2 または 3 に記載の物流情報管理装置。

【請求項 5】 上記署名鍵使用限度情報保持手段は、上記署名モジュール内にあることを特徴とする請求項 1、2、3 または 4 に記載の物流情報管理装置。

【請求項 6】 上記物流情報処理モジュールは、
情報検証モジュールと、
情報生成モジュールと、
から構成され、

上記情報検証モジュールは、

上記データキャリアのデータを読み出す第一の読み取り手段と、

上記データキャリアから読み出した情報を処理する第一の情報検証部と、
から構成され、

上記情報生成モジュールは、

上記データキャリアのデータを読み出す第二の読み取り手段と、

上記データキャリアに情報を書込む書き込み手段と、

上記データキャリアへ書込む情報を処理する情報生成部と、
から構成される

ことを特徴とする請求項 1、2、3、4 または 5 に記載の物流情報管理装置。

【請求項 7】 上記物流情報管理モジュールは、
第二の情報検証モジュールと、
署名鍵情報生成モジュールと、
から構成され、

上記第二の情報検証モジュールは、

物流情報検証部と、
第二の通信手段と、
から構成され、
上記署名鍵情報生成モジュールは、
署名鍵生成部と、
第三の通信手段と、
から構成される

ことを特徴とする請求項 1、2、3、4、5 または 6 に記載の物流情報管理装置。

【請求項 8】 上記第一の検証鍵記憶手段と上記第二の検証鍵記憶手段に記憶される検証鍵は、全ての物流情報処理モジュールと物流情報管理モジュールとで共通であることを特徴とする請求項 1～7 のいずれかに記載の物流情報管理装置。

【請求項 9】 第一の情報検証手段と第二の情報検証手段は、同じ動作をすることを特徴とする請求項 1～8 のいずれかに記載の物流情報管理装置。

【請求項 10】 第一の情報検証部は、第一の情報検証手段が使用する検証鍵を選択する第一の検証鍵選択手段を持つことを特徴とする請求項 1～8 のいずれかに記載の物流情報管理装置。

【請求項 11】 第二の情報検証部は、第二の情報検証手段が使用する検証鍵を選択する第二の検証鍵選択手段を持つことを特徴とする請求項 9 または 10 に記載の物流情報管理装置。

【請求項 12】 署名鍵情報生成部は、署名鍵を選択する署名鍵選択手段を持つことを特徴とする請求項 1～11 のいずれかに記載の物流情報管理装置。

【請求項 13】 データキャリアに格納される情報は少なくとも、製品識別子と、署名者識別子と、受領者識別子と、署名値とを含む情報を一単位として格納されることを特徴とする請求項 1～12 のいずれかに記載の物流情報管理装置。

【請求項 14】 データキャリアに格納される情報は少なくとも、検証鍵識別子とを含む情報を一単位として格納されることを特徴とする請求項 13 の物流

情報管理装置。

【請求項 1 5】 データキャリアに格納される情報は少なくとも、物流情報管理モジュール識別子とを含む情報を一単位として格納されることを特徴とする請求項 1 1、1 2、1 3、または 1 4 に記載の物流情報管理装置。

【請求項 1 6】 データキャリアに格納される情報は少なくとも、製品識別子と、署名者識別子と、受領者識別子とを含む情報を一単位として格納され、上記単位毎の情報とは別に署名値を持つことを特徴とする請求項 1 ～ 1 2 のいずれかに記載の物流情報管理装置。

【請求項 1 7】 データキャリアに格納される情報は少なくとも、製品識別子と、署名者識別子と、受領者識別子と、検証鍵識別子とを含む情報を一単位として格納され、検証鍵識別子毎に検証鍵識別子と対応づけられた署名値を持つことを特徴とする請求項 1 ～ 1 2 のいずれかに記載の物流情報管理装置。

【請求項 1 8】 物品に添付され、上記物品に関する情報を記憶するデータキャリアにおいて、

上記物品の流通過程における 1 またはひとまとまりの取引ごとに生成される上記物品の流通情報と、1 の上記流通情報の少なくとも一部または一連の上記流通情報のおおのの少なくとも一部の署名値とを記憶することを特徴とするデータキャリア。

【請求項 1 9】 上記物品の流通情報は、上記物品の識別子と、上記物品を受領した者の識別子と、上記署名値を生成する者の識別子とを少なくとも含む請求項 1 8 記載のデータキャリア。

【請求項 2 0】 物品に添付され、上記物品に関する情報を記憶するデータキャリアに情報を読み書きするとともに、上記物品の流通に関する情報を管理する物流情報管理モジュールとの間で情報をやり取りして上記物品に関する情報を処理する物流情報処理モジュールにおいて、

上記データキャリアのデータを読み出す読み取り手段と、

上記データキャリアに情報を書込む書き込み手段と、

上記データキャリアから読み出した情報を検証する第一の情報検証部と、

上記データキャリアへ書込む情報を処理する情報生成部と、

上記物流情報管理モジュールとの通信を行うための第一の通信手段と、
から構成され、

上記第一の情報検証部は、

上記データキャリアから読み出した情報の検証を行う第一の情報検証手段と

、
上記第一の情報検証手段が情報の検証に用いる検証鍵を記憶する第一の検証
鍵記憶手段と、

から構成され、

上記情報生成部は、

上記データキャリアに書込む情報を生成する物流情報生成手段と、

署名生成処理を行う署名モジュールと、

上記署名モジュールが電子署名生成時に使用する署名鍵情報を記憶する署名
鍵記憶手段と、

上記署名鍵記憶手段に記憶された署名鍵情報を選択するための署名鍵情報選
択手段と、

上記署名鍵情報を物流情報管理モジュールから取得するための署名鍵情報取
得手段と、

から構成される、

上記署名モジュールは、

上記物流情報生成手段によって生成された情報に対する電子署名を生成す
る署名手段と、

上記署名手段が電子署名生成時に使用する署名者秘密情報を記憶する第一
の署名者秘密情報記憶手段と、

から構成されること特徴とする物流情報処理モジュール。

【請求項 2 1】 物品に添付され、上記物品に関する情報を記憶するデータ
キャリアに情報を読み書きするとともに上記データキャリアの情報を検証するた
めの物流情報処理モジュールとの間で情報をやり取りし、上記物品の流通に関す
る情報を管理する物流情報管理モジュールにおいて、

上記物流情報処理モジュールとの通信を行うための通信手段と、

上記物流情報処理モジュールから送られてきた情報を処理するための情報検証部と、

上記物流情報処理モジュールへ送る署名鍵情報を処理するための署名鍵情報生成部と、
から構成され、

上記情報検証部は、

上記物流情報処理モジュールから送られてきた情報を検証するための情報検証手段と、

上記情報検証手段が情報の検証に用いる検証鍵を記憶する検証鍵記憶手段と

から構成され、

上記署名鍵情報生成部は、

上記物流情報処理モジュールが物流情報生成時に用いる署名鍵情報を生成する署名鍵情報生成手段と、

上記署名鍵情報生成手段が署名鍵情報生成時に用いる署名鍵を記憶する署名鍵記憶手段と、

上記署名鍵情報生成手段が署名鍵情報生成時に用いる署名者秘密情報を選択するための署名者秘密情報選択手段と、

上記署名者秘密情報を記憶する署名者秘密情報記憶手段と、

から構成される

ことを特徴とする物流情報管理モジュール。

【請求項 22】 物品に添付され、上記物品に関する情報を記憶するデータキャリアと、上記データキャリアに情報を読み書きするための物流情報処理モジュールと、上記物品の流通に関する情報を管理する物流情報管理モジュールとを用いて物流情報を管理する方法において、

上記データキャリアのデータを読み出す読み取りステップと、

上記データキャリアに情報を書込む書き込みステップと、

上記データキャリアから読み出した情報を検証する第一の情報検証ステップと

上記データキャリアへ書込む情報を処理する情報生成ステップと、
上記物流情報管理モジュールとの通信を行うための第一の通信ステップと、
から構成され、

上記第一の情報検証ステップは、

上記データキャリアから読み出した情報の検証を行う第一の情報検証サブステップと、

上記第一の情報検証サブステップにおいて情報の検証に用いる検証鍵を記憶する第一の検証鍵記憶サブステップと、

から構成され、

上記情報生成ステップは、

上記データキャリアに書込む情報を生成する物流情報生成サブステップと、
署名生成処理を行う署名サブステップと、

上記署名サブステップにおいて電子署名生成時に使用する署名鍵情報を記憶する署名鍵記憶ステップと、

上記署名鍵記憶サブステップにおいて記憶された署名鍵情報を選択するための署名鍵情報選択ステップと、

上記署名鍵情報を物流情報管理モジュールから取得するための署名鍵情報取得サブステップと、

から構成され、

上記署名サブステップは、

上記物流情報生成モジュールによって生成された情報に対する電子署名を生成する署名マイクロステップと、

上記署名マイクロステップにおいて電子署名生成時に使用する署名者秘密情報を記憶する第一の署名者秘密情報記憶マイクロステップと、

から構成されることを特徴とする物流情報処理方法。

【請求項 23】 物品に添付され、上記物品に関する情報を記憶するデータキャリアと、上記データキャリアに情報を読み書きするための物流情報処理モジュールと、上記物品の流通に関する情報を管理する物流情報管理モジュールとを用いて物流情報を管理する方法において、

上記物流情報処理モジュールとの通信を行うための通信ステップと、

上記物流情報処理モジュールから送られてきた情報を処理するための情報検証ステップ部と、

上記物流情報処理モジュールへ送る署名鍵情報を処理するための署名鍵情報生成ステップと、
から構成され、

上記情報検証ステップは、

上記物流情報処理モジュールから送られてきた情報を検証するための情報検証サブステップと、

上記情報検証サブステップにおいて情報の検証に用いる検証鍵を記憶する検証鍵記憶サブステップと、

から構成され、

上記署名鍵情報生成ステップは、

上記物流情報処理モジュールが物流情報生成時に用いる署名鍵情報を生成する署名鍵情報生成サブステップと、

上記署名鍵情報生成サブステップにおいて署名鍵情報生成時に用いる署名鍵を記憶する署名鍵記憶サブステップと、

上記署名鍵情報生成サブステップにおいて署名鍵情報生成時に用いる署名者秘密情報を選択するための署名者秘密情報選択サブステップと、

上記署名者秘密情報を記憶する署名者秘密情報記憶サブステップと、

から構成される

ことを特徴とする物流情報管理方法。

【請求項 2 4】 物品に添付され、上記物品に関する情報を記憶するデータキャリアと、上記データキャリアに情報を読み書きするための物流情報処理モジュールと、上記物品の流通に関する情報を管理する物流情報管理モジュールとを用いて物流情報を管理するために用いられる物流情報処理用のコンピュータプログラム製品において、

上記データキャリアのデータを読み出す読み取りステップと、

上記データキャリアに情報を書込む書き込みステップと、

上記データキャリアから読み出した情報を検証する第一の情報検証ステップと

上記データキャリアへ書込む情報を処理する情報生成ステップと、

上記物流情報管理モジュールとの通信を行うための第一の通信ステップと、

をコンピュータシステムに実行させるために用いられ、

上記第一の情報検証ステップは、

上記データキャリアから読み出した情報の検証を行う第一の情報検証サブステップと、

上記第一の情報検証サブステップにおいて情報の検証に用いる検証鍵を記憶する第一の検証鍵記憶サブステップと、

から構成され、

上記情報生成ステップは、

上記データキャリアに書込む情報を生成する物流情報生成サブステップと、

署名生成処理を行う署名サブステップと、

上記署名サブステップにおいて電子署名生成時に使用する署名鍵情報を記憶する署名鍵記憶ステップと、

上記署名鍵記憶サブステップにおいて記憶された署名鍵情報を選択するための署名鍵情報選択ステップと、

上記署名鍵情報を物流情報管理モジュールから取得するための署名鍵情報取得サブステップと、

から構成され、

上記署名サブステップは、

上記物流情報生成モジュールによって生成された情報に対する電子署名を生成する署名マイクロステップと、

上記署名マイクロステップにおいて電子署名生成時に使用する署名者秘密情報を記憶する第一の署名者秘密情報記憶マイクロステップと、

から構成されることを特徴とする物流情報処理用コンピュータプログラム製品。

【請求項 25】 物品に添付され、上記物品に関する情報を記憶するデータ

キャリアと、上記データキャリアに情報を読み書きするための物流情報処理モジュールと、上記物品の流通に関する情報を管理する物流情報管理モジュールとを用いて物流情報を管理するために用いられるコンピュータプログラム製品において、

上記物流情報処理モジュールとの通信を行うための通信ステップと、

上記物流情報処理モジュールから送られてきた情報を処理するための情報検証ステップ部と、

上記物流情報処理モジュールへ送る署名鍵情報を処理するための署名鍵情報生成ステップと、

をコンピュータシステムに実行させるために用いられ、さらに、

上記情報検証ステップは、

上記物流情報処理モジュールから送られてきた情報を検証するための情報検証サブステップと、

上記情報検証サブステップにおいて情報の検証に用いる検証鍵を記憶する検証鍵記憶サブステップと、

から構成され、

上記署名鍵情報生成ステップは、

上記物流情報処理モジュールが物流情報生成時に用いる署名鍵情報を生成する署名鍵情報生成サブステップと、

上記署名鍵情報生成サブステップにおいて署名鍵情報生成時に用いる署名鍵を記憶する署名鍵記憶サブステップと、

上記署名鍵情報生成サブステップにおいて署名鍵情報生成時に用いる署名者秘密情報を選択するための署名者秘密情報選択サブステップと、

上記署名者秘密情報を記憶する署名者秘密情報記憶サブステップと、

から構成される

ことを特徴とする物流情報管理用コンピュータプログラム製品。

【請求項 2 6】 物品に添付され、上記物品に関する情報を記憶するデータキャリアと、上記データキャリアに情報を読み書きするための物流情報処理モジュールと、上記物品の流通に関する情報を管理する物流情報管理モジュールとを

含んで構成されている物流情報管理装置において、

上記物流情報処理モジュールは、

上記データキャリアのデータを読み出す読み取り手段と、

上記データキャリアに情報を書込む書き込み手段と、

上記データキャリアから読み出した情報を検証する第一の情報検証部と、

上記データキャリアへ書込む情報を処理する情報生成部と、

上記物流情報管理モジュールとの通信を行うための第一の通信手段と、

から構成され、

上記第一の情報検証部は、

上記データキャリアから読み出した情報の検証を行う情報検証手段と、

上記第一の情報検証手段が情報の検証に用いる検証鍵を記憶する検証鍵記憶手段と、

から構成され、

上記情報生成部は、

上記データキャリアに書込む情報を生成する物流情報生成手段と、

署名生成処理を行う署名モジュールと、

上記署名モジュールが電子署名生成時に使用する署名鍵情報を記憶する署名鍵記憶手段と、

上記署名鍵記憶手段に記憶された署名鍵情報を選択するための署名鍵情報選択手段と、

上記署名鍵情報を物流情報管理モジュールから取得するための署名鍵情報取得手段と、

から構成され、

上記署名モジュールは、

上記物流情報生成手段によって生成された情報に対する電子署名を生成する署名手段と、

上記署名手段が電子署名生成時に使用する署名者秘密情報を記憶する第一の署名者秘密情報記憶手段と、

から構成され、

上記物流情報管理モジュールは、
上記物流情報処理モジュールとの通信を行うための第二の通信手段と、
上記物流情報処理モジュールへ送る署名鍵情報を処理するための署名鍵情報生成部と、
から構成され、

上記署名鍵情報生成部は、
上記物流情報処理モジュールが物流情報生成時に用いる署名鍵情報を生成する署名鍵情報生成手段と、

上記署名鍵情報生成モジュールが署名鍵情報生成時に用いる署名鍵を記憶する署名鍵記憶手段と、

上記署名鍵情報生成モジュールが署名鍵情報生成時に用いる署名者秘密情報を選択するための署名者秘密情報選択手段と、

上記署名者秘密情報を記憶する第二の署名者秘密情報記憶手段と、
から構成される
ことを特徴とする物流情報管理装置。

【請求項 27】 物品に添付され、上記物品に関する情報を記憶するデータキャリアと、上記データキャリアに情報を読み書きするための物流情報処理モジュールと、上記物品の流通に関する情報を管理する物流情報管理モジュールとを含んで構成されている物流情報管理装置において、

上記物流情報処理モジュールは、
上記データキャリアに情報を書込む書き込み手段と、
上記データキャリアへ書込む情報を処理する情報生成部と、
上記物流情報管理モジュールとの通信を行うための第一の通信手段と、
から構成され、

上記情報生成部は、
上記データキャリアに書込む情報を生成する物流情報生成手段と、
署名生成処理を行う署名モジュールと、
上記署名モジュールが電子署名生成時に使用する署名鍵情報を記憶する署名鍵記憶手段と、

上記署名鍵記憶手段に記憶された署名鍵情報を選択するための署名鍵情報選択手段と、

上記署名鍵情報を物流情報管理モジュールから取得するための署名鍵情報取得手段と、

から構成され、

上記署名モジュールは、

上記物流情報生成手段によって生成された情報に対する電子署名を生成する署名手段と、

上記署名手段が電子署名生成時に使用する署名者秘密情報を記憶する第一の署名者秘密情報記憶手段と、

から構成され、

上記物流情報管理モジュールは、

上記物流情報処理モジュールとの通信を行うための第二の通信手段と、

上記物流情報処理モジュールから送られてきた情報を処理するための情報検証部と、

上記物流情報処理モジュールへ送る署名鍵情報を処理するための署名鍵情報生成部と、

から構成され、

上記情報検証部は、

上記物流情報処理モジュールから送られてきた情報を検証するための情報検証手段と、

上記情報検証手段が情報の検証に用いる検証鍵を記憶する検証鍵記憶手段と

、

から構成され、

上記署名鍵情報生成部は、

上記物流情報処理モジュールが物流情報生成時に用いる署名鍵情報を生成する署名鍵情報生成手段と、

上記署名鍵情報生成手段が署名鍵情報生成時に用いる署名鍵を記憶する署名鍵記憶手段と、

上記署名鍵情報生成手段が署名鍵情報生成時に用いる署名者秘密情報を選択するための署名者秘密情報選択手段と、

上記署名者秘密情報を記憶する第二の署名者秘密情報記憶手段と、
から構成される

ことを特徴とする物流情報管理装置。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、物品にデータキャリアを添付し、データキャリア内の情報を利用して物品の流通を管理する技術に関する。

【0 0 0 2】

【従来の技術】

従来の流通管理システムとしては、商品の管理に電磁的に読み書き可能なデータキャリアを使用することで、流通過程での作業の効率や確実性を向上させようとするものがある。

【0 0 0 3】

例えば、特開平 5－2 9 8 3 3 2 号公報の「データキャリアを用いた物流システム」には、非接触で情報の読み書きが可能なデータキャリアを用いた流通システムが記述されている。データキャリアには商品固有のデータが書き込まれており、例えば、品名、産地、個数などが記載されている。商品が競り落とされた場合、データキャリアには仲買人の ID が追加記載される。仲買人が商品を小売りに売の場合はさらに小売り人の ID、個数、値段、発送日時などが追加記載される。また、これら商品を乗せる台車やトラックにもデータキャリアが設けられており、積み荷の個数など台車に積まれた商品全体の情報が予め書かれている。商品は小売り人の ID により自動的にそれぞれの台車に振り分けられ、さらに同 ID の記載されたトラックに積み込まれる。

【0 0 0 4】

このような方法を用いることで、商品の発送先、発送日時、発送個数などの間違いを回避することができる。更にまたデータキャリアを市場への出入りチェッ

クに用いることで商品の盗難防止を図ることができる。

【 0 0 0 5 】

また、特開平 1 0 - 3 2 4 4 0 5 号公報の「商品配送システム」には、電子荷札により商品の内容物の確認を正確に行うことができる商品配送システムが記述されている。商品には予め電子荷札が取り付けられており、配送側において、電子荷札から無線により商品情報を読み取って、梱包物内の商品を示す内容物情報を作成し、受け取り側に渡す。受け取り側では、同様に電子荷札から無線により商品情報を受け取って、内容物情報と比較する事で検品を行う。

【 0 0 0 6 】

このような商品配送システムを用いれば、各商品に取り付けられた電子荷札により梱包後に内容物の確認を行い、また受け取り側において梱包した状態で電子荷札から読み取った商品情報と商品のリストを示す内容物情報とを比較して検品を行うので、人間を介在させずに検品を行う事ができ、商品の配送およびその確認を正確に行う事ができる。

【 0 0 0 7 】

これら流通管理システムの改良により商品配送管理の正確性が向上してきたが、今なお存在する問題として、ブランド品などにおいて商品の流通過程で偽物が混入するという問題がある。上記に述べた方法によれば、確かに配送作業を自動化でき商品の検品を正確に行う事ができるが、流通業者が不正を行うことでその流通過程に偽物を混入させることは容易であり、また偽物が混入した場合にどの流通業者が混入したのかをつきとめることは非常に困難であった。

【 0 0 0 8 】

このような不正を排除するために、上記電子荷札（データキャリア）に情報を書き込む際に署名を付けて書き込むという改善策が考えられる。以後の説明では電子荷札をデータキャリアと呼びかえる。データキャリアには各流通業者の取り扱い記録とそれに対する署名が書き込まれているので、偽物が発見された場合は、流通経路を特定することで偽物を混入した業者をある程度特定することが可能である。また、流通業者の取り扱い情報の署名が付加されていなかったり不正だった場合は、その業者またはそこから商品を受け取った業者が怪しいということ

になる。

【0009】

このような流通情報に対して署名を付ける技術としては、デジタル署名技術が挙げられる。デジタル署名技術の代表的な例として、離散対数問題の困難性に安全性の根拠をおくエルガマル署名法がある。以下、エルガマル署名法について説明する。

【0010】

署名鍵を (x, α, p) 、検証鍵を (y, α, p) とする。ここで p は素数、 α は p 未満のある正定数である。これらの整数の間には、式 (1) で示す関係がある。

【0011】

【数1】

$$(1) \quad y = \alpha^x \pmod{p}$$

公開整数 y から秘密整数 x を求めるのは離散対数問題であり、 p が十分大きければ (500ビット以上) x を計算により求めることは困難である。

【0012】

証明者は $p-1$ と互いに素な乱数 k を生成し、式 (2) (3) により、メッセージ m に対する署名 (r, s) を計算する。

【0013】

【数2】

$$(2) \quad r = \alpha^k \pmod{p}$$

【0014】

【数3】

$$(3) \quad s = (h(m) - xr) k^{-1} \pmod{p-1}$$

ここで、 h は一方向性ハッシュ関数を表す。証明者はメッセージ m と署名 (r, s) を検証側に送る。

【0015】

検証者は m 、 (r, s) を受け取って、式 (4) が成立するかを確認する。

【0016】

【数4】

$$(4) \quad \alpha^{h(m)} = y^r r^s \pmod{p}$$

式(4)が成立することで、mが証明者により作成されたメッセージであることが証明される。

【0017】

デジタル署名法にはこの他に、同様に離散対数問題の困難さに安全性の根拠を置くDSA (Digital Signature Algorithm)、Schnorr署名法や、零知識証明法に基づいたG-Q (Guillou, Quisquater) および、一般的によく知られるRSA (Rivest, Shamir, Adleman) 署名法等が挙げられる。

【0018】

【発明が解決しようとする課題】

しかし、以上で説明したようなデジタル署名法を、上記データキャリアに書き込む流通情報などの署名に用いた場合は、次のような課題がある。

【0019】

デジタル署名法を用いる場合は、流通業者各々の署名鍵に対する検証鍵の証明書 (Certificate) を発行するための認証局 (Certificate Authority) が必要となる。このような認証局を全国または世界中の流通業者をサポートできるような規模にしようとすると膨大な設備と費用がかかるという課題がある。また、流通情報に対する署名を検証する場合、検証者は署名をした流通業者各々に対して、証明書が必要になり、検証者は認証を行う証明者に対する証明書をその都度認証局から入手するか、予めテーブルに保管しておく必要がある。証明書をその都度入手する方法は、署名者が複数に及ぶ場合は検証に時間がかかるという課題があり、テーブルに保管する場合は、証明書の有効期限の管理や無効化された場合のチェック等の複雑な管理が必要となる。また署名鍵は機密情報なのでその管理にセキュリティ上の注意が必要となる。このようにデジタル署名法を用いると、認証局が必要となり、また証明書の管理や流通業者でのセキュリティの確保等、多大な設備や手間がかかるという課題がある。

【0 0 2 0】

また、一般的にデジタル署名を利用する場合は、署名鍵は署名者が管理するので署名回数を制限することができない。その場合、同一商品に対して複数回の署名を施すことが可能であり、署名を行う流通業者が偽造商品に本物の商品と同一のIDを振り、別の受け取り業者へ配送するということが可能である。このとき商品に付けられた署名は検証に成功するので、同一のIDを持つ商品の存在が発覚するまで偽造商品が本物として出回ることになってしまうという問題がある。

【0 0 2 1】

本発明は、上述の事情を考慮してなされたものであり、多大な設備や手間を必要とすることなく、署名を行え、しかも、署名の回数を制限することができる物流流通管理技術を提供することを目的としている。

【0 0 2 2】

【課題を解決するための手段】

本発明の物流情報管理装置は、上述の目的を達成するために、物品に添付され、物品に関する情報を記憶するデータキャリアと；データキャリアの情報を読み書きするための物流情報処理モジュールと；物品の流通に関する情報を管理する物流情報管理モジュールとから構成される。

【0 0 2 3】

上記物流情報処理モジュールは：上記データキャリアのデータを読み出す読み取り手段と；上記データキャリアに情報を書込む書き込み手段と；上記データキャリアから読み出した情報を検証する第一の情報検証部と；上記データキャリアへ書込む情報を処理する情報生成部と；上記物流情報管理モジュールとの通信を行うための第一の通信手段とから構成されている。

【0 0 2 4】

そして上記第一の情報検証部は：上記データキャリアから読み出した情報の検証を行う第一の情報検証手段と；上記第一の情報検証手段が情報の検証に用いる検証鍵を記憶する第一の検証鍵記憶手段とから構成されている。

【0 0 2 5】

また、上記情報生成部は：上記データキャリアに書込む情報を生成する物流情報生成手段と；署名生成処理を行う署名モジュールと；上記署名モジュールが電子署名生成時に使用する署名鍵情報を記憶する署名鍵記憶手段と；上記署名鍵記憶手段に記憶された署名鍵情報を選択するための署名鍵情報選択手段と；上記署名鍵情報を上記物流情報管理モジュールから取得するための署名鍵情報取得手段とから構成されている。

【 0 0 2 6 】

上記署名モジュールは：上記物流情報生成手段によって生成された情報に対する電子署名を生成する署名手段と；上記署名手段が電子署名生成時に使用する署名者秘密情報を記憶する第一の署名者秘密情報記憶手段とから構成されている。

【 0 0 2 7 】

また、上記物流情報管理モジュールは：上記物流情報処理モジュールとの通信を行うための第二の通信手段と；上記物流情報処理モジュールから送られてきた情報を処理するための第二の情報検証部と；上記物流情報処理モジュールへ送る署名鍵情報を処理するための署名鍵情報生成部とから構成されている。

【 0 0 2 8 】

上記第二の情報検証部は：上記物流情報処理モジュールから送られてきた情報を検証するための第二の情報検証手段と；上記第二の情報検証手段が情報の検証に用いる検証鍵を記憶する第二の検証鍵記憶手段とから構成されている。

【 0 0 2 9 】

上記署名鍵情報生成部は：上記物流情報処理モジュールが物流情報生成時に用いる署名鍵情報を生成する署名鍵情報生成手段と；上記署名鍵情報生成手段が署名鍵情報生成時に用いる署名鍵を記憶する署名鍵記憶手段と；上記署名鍵情報生成手段が署名鍵情報生成時に用いる署名者秘密情報を選択するための署名者秘密情報選択手段と；上記署名者秘密情報を記憶する第二の署名者秘密情報記憶手段とから構成されている。

【 0 0 3 0 】

このような構成によれば、物流情報処理モジュールは、物流情報管理モジュールからモジュール固有の署名鍵情報を取得し、この署名鍵情報と署名書秘密情報

とを用いて物流情報に署名を行い、この署名の検証には通有性のある検証鍵を用いるようになっている。したがって、署名者ごとの検証鍵をやり取りする必要がなく、設備や手間を少なくすることができる。

【0031】

また、本発明の物流情報管理装置において、署名モジュールは、物流情報処理モジュールから取り外し交換することができるようにしてもよい。

【0032】

また、本発明の物流情報管理装置において、署名モジュールは、耐タンパー性を持つようにしてよい。

【0033】

また、本発明の物流情報管理装置において、情報生成部は、署名鍵使用限度情報保持手段を持ち、署名鍵情報選択手段は、特定回数以上署名に用いた署名鍵情報は選択されないようにしてもよい。

【0034】

また、本発明の物流情報管理装置において、署名鍵使用限度情報保持手段は、署名モジュール内に存在するようにしてもよい。

【0035】

また、本発明の物流情報管理装置において、物流情報処理モジュールは、情報検証モジュールと、情報生成モジュールとから構成され、情報検証モジュールは、データキャリアのデータを読み出す第一の読み出し手段と、データキャリアから読み出した情報を処理する第一の情報検証部とから構成され、情報生成モジュールは、データキャリアのデータを読み出す第二の読み出し手段と、データキャリアに情報を書込む書き込み手段と、データキャリアへ書込む情報を処理する情報生成部とから構成されるようにしてもよい。

【0036】

また、本発明の物流情報管理装置において、物流情報管理モジュールは、第二の情報検証モジュールと、署名鍵情報生成モジュールとから構成され、第二の情報検証モジュールは、物流情報検証部と、第二の通信手段とから構成され、署名鍵情報生成モジュールは、署名鍵生成部と、第三の通信手段とから構成されるよ

うにしてもよい。

【 0 0 3 7 】

また、本発明の物流情報管理装置において、第一の検証鍵記憶手段と第二の検証鍵記憶手段に記憶される検証鍵は、全ての物流情報処理モジュールと物流情報管理モジュールとで共通としてもよい。

【 0 0 3 8 】

また、本発明の物流情報管理装置において、第一の情報検証手段と第二の情報検証手段とは、同じ動作をするようにしてもよい。

【 0 0 3 9 】

また、本発明の物流情報管理装置において、第一の情報検証部は、第一の情報検証手段が使用する検証鍵を選択する第一の検証鍵選択手段を持つようにしてもよい。（物流情報管理モジュールまたは物流管理センタごとに検証鍵を変えることができる。例えば、メーカー別のセンタがあって、仲介業者や小売店は異なるメーカーの製品を扱うような場合が考えられる。公開鍵 I D で切り替える方法とセンタ I D で切り替える方法が考えられる。）

また、本発明の物流情報管理装置において、第二の情報検証部は、第二の情報検証手段が使用する検証鍵を選択する第二の検証鍵選択手段を持つようにしてもよい。

【 0 0 4 0 】

また、本発明の物流情報管理装置において、署名鍵情報生成部は、署名鍵を選択する署名鍵選択手段を持つようにしてもよい。

【 0 0 4 1 】

また、本発明の物流情報管理装置において、データキャリアに格納される情報は少なくとも、製品識別子と、署名者識別子と、受領者識別子と、署名値とを含む情報を一単位として格納されるようにしてもよい。

【 0 0 4 2 】

また、本発明の物流情報管理装置において、データキャリアに格納される情報は少なくとも、検証鍵識別子とを含む情報を一単位として格納されるようにしてもよい。

【0043】

また、本発明の物流情報管理装置において、データキャリアに格納される情報は少なくとも、物流情報管理モジュール識別子と、を含む情報を一単位として格納されるようにしてもよい。

また、本発明の物流情報管理装置において、データキャリアに格納される情報は少なくとも、製品識別子と、署名者識別子と、受領者識別子とを含む情報を一単位として格納され、上記単位毎の情報とは別に署名値を持つようにしてもよい。

【0044】

また、本発明の物流情報管理装置において、データキャリアに格納される情報は少なくとも、製品識別子と、署名者識別子と、受領者識別子と、検証鍵識別子とを含む情報を一単位として格納され、検証鍵識別子毎に検証鍵識別子と対応づけられた署名値を持つようにしてもよい。

【0045】

また、本発明によれば、物品に添付され、上記物品に関する情報を記憶するデータキャリアに、上記物品の流通過程における1またはひとまとまりの取引ごとに生成される上記物品の流通情報（例えば取り扱い記録）と、1の上記流通情報の少なくとも一部または一連の上記流通情報のおおのの少なくとも一部の署名値とを記憶するようにしている。

【0046】

このようなデータキャリアを用いることにより偽物の混入等の不正を排除することができる。

【0047】

上記物品の流通情報は、上記物品の識別子と、上記物品を受領した者の識別子と、上記署名値を生成する者の識別子とを少なくとも含むようにすることができる。

【0048】

より具体的には、上記流通情報は少なくとも、製品識別子と、署名者識別子と、受領者識別子と、署名値とを含む情報を一単位として格納されるようにしてもよいし、少なくとも、検証鍵識別子とを含む情報を一単位として格納されるよう

にしてもよいし、少なくとも、物流情報管理モジュール識別子と、を含む情報を一単位として格納されるようにしてもよいし、少なくとも、製品識別子と、署名者識別子と、受領者識別子とを含む情報を一単位として格納され、上記単位毎の情報とは別に署名値を持つようにしてもよいし、さらに、少なくとも、製品識別子と、署名者識別子と、受領者識別子と、検証鍵識別子とを含む情報を一単位として格納され、検証鍵識別子毎に検証鍵識別子と対応づけられた署名値を持つようにしてもよい。

【0049】

また、本発明において、物流情報処理モジュールまたは物流情報管理モジュールの一方でのみ、署名の検証を行うようにしてもよい。

【0050】

また、物流情報処理モジュールを情報読み取りおよび検証を行う部分と、情報を書き込み署名する部分とに分離しても良い。さらに物流情報管理モジュールを検証を行う部分と、署名鍵情報を管理し、物流情報処理モジュールに送る部分とに分離しても良い。

【0051】

また、本発明は方法の形態で実現されても良く、また、少なくとも、一部が、コンピュータプログラム製品として実現されても良い。

【0052】

【発明の実施の態様】

以下、本発明を実施例を用いて詳細に説明する。

【0053】

〔実施例1〕

図1は、本実施例における構成を示している。図1において、本実施例の物流管理システムは、物品に添付され、物品に関する情報を記憶するデータキャリア1と、データキャリア1の情報を読み書きするための物流情報処理モジュール2と、物品の流通に関する情報を管理する物流情報管理モジュール3とから構成されている。

【0054】

物流情報処理モジュール 2 は、データキャリア 1 のデータを読み出す読み取り手段 4 と、データキャリア 1 に情報を書込む書き込み手段 5 と、データキャリア 1 から読み出した情報を検証する第一の情報検証部 6 と、データキャリア 1 へ書込む情報を処理する情報生成部 7 と、物流情報管理モジュール 3 との通信を行うための第一の通信手段 8 とから構成されている。第一の情報検証部 6 は、データキャリア 1 から読み出した情報の検証を行う第一の情報検証手段 9 と、第一の情報検証手段 9 が情報の検証に用いる検証鍵を記憶する第一の検証鍵記憶手段 10 とから構成されている。情報生成部 7 は、データキャリア 1 に書込む情報を生成する物流情報生成手段 11 と、署名生成処理を行う署名モジュール 12 と、署名生成に用いる署名鍵情報を選択するための署名鍵情報選択手段 13 と、署名手段 16（署名モジュール 12）が電子署名生成時に使用する署名鍵情報を記憶する署名鍵情報記憶手段 14 と、署名鍵情報を物流情報管理モジュール 3 から取得するための署名鍵情報取得手段 15 とから構成されている。署名モジュール 12 は、物流情報生成手段 11 によって生成された情報に対する電子署名を生成する署名手段 16 と、署名手段 16 が電子署名生成時に使用する署名者秘密情報を記憶する第一の署名者秘密情報記憶手段 17 とから構成されている。

【0055】

物流情報管理モジュール 3 は、物流情報処理モジュール 2 との通信を行うための第二の通信手段 18 と、物流情報処理モジュール 2 から送られてきた情報を処理するための第二の情報検証部 19 と、物流情報処理モジュール 2 へ送る署名鍵情報を処理するための署名鍵情報生成部 20 とから構成されている。第二の情報検証部 19 は、物流情報処理モジュール 2 から送られてきた情報を検証するための第二の情報検証手段 21 と、第二の情報検証手段 21 が情報の検証に用いる検証鍵を記憶する第二の検証鍵記憶手段 22 とから構成されている。署名鍵情報生成部 20 は、物流情報処理モジュール 2 が物流情報生成時に用いる署名鍵情報を生成する署名鍵情報生成手段 23 と、署名鍵情報生成手段 23 が署名鍵情報生成時に用いる署名鍵を記憶する署名鍵記憶手段 24 と、署名鍵情報生成手段 23 が署名鍵情報生成時に用いる署名者秘密情報を選択するための署名者秘密情報選択手段 25 と、署名者秘密情報を記憶する第二の署名者秘密情報記憶手段 26 とか

ら構成される。

【0056】

図2は、本実施例においてデータキャリア1に記憶される物流情報の例を示している。データキャリア1には被署名部と署名値、必要ならその他の情報を一つのレコードとして物流情報が記憶されており、被署名部は少なくとも製品識別子、署名者識別子、受領者識別子を含んでいる。

【0057】

図3は、データキャリア1から読み出した図2の物流情報を処理する場合の第一の情報検証部6での物流情報検証処理の流れを示している。

【0058】

図3において、読み取り手段4によりデータキャリア1から読み出された物流情報が送られてくると、第一の情報検証手段9は、受領者識別子用の変数 `rid` をクリアする（ステップS301）。第一の署名検証鍵記憶手段10から署名検証鍵を読み出し、変数 `E`、変数 `n` にセットする（ステップS302）。物流情報から最初のレコードを読み出す（ステップS303）。読み出したレコードから被署名部を読み出し、変数 `data` にセットする（ステップS304）。`data` から製品識別子を読み出し、変数 `pid` にセットする（ステップS305）。`data` から署名者識別子を読み出し、変数 `sid` にセットする（ステップS306）。関数 $H(data, pid, sid)$ によってハッシュ値を計算し、変数 `h` にセットする（ステップS307）。ここでハッシュ関数 $H()$ は引数 `data`、`pid`、`sid` から一意の値を返す関数 $F(data, pid, sid)$ 、例えば、`data | pid | sid`（`|` はビット列の連結）、の返す値に対してSHA-1、MD5などハッシュ関数を適用したものであり、関数 $F()$ 、ハッシュ関数は特定のものに限定されない。ステップS303で読み出したレコードから署名部を読み出し、変数 `sign` にセットする（ステップS308）。検証用の値を次式によって計算し、変数 `val` にセットする（ステップS309）。

【0059】

【数5】

$\text{sign}^E \bmod n$ (\bmod は剰余演算)

ここで、ステップ S309 は署名アルゴリズムとして RSA 署名を用いている。ステップ S309 で他の署名アルゴリズムを用いる場合は、第一の署名検証鍵記憶手段 10 に記憶され、ステップ S308 で読み込まれる鍵はそのアルゴリズムに適したものが使用される。ステップ S307 で計算したハッシュ値 h とステップ S309 で計算した値 val が等しいか確認する。等しくなければ、署名がおかしい旨のエラーを返し、終了し、等しければ、ステップ S311 に進む (ステップ S310)。 rid に値がセットされているか確認し (ステップ S311)、 rid に値がセットされていないければ、ステップ S313 へ進み、 rid に値がセットされていれば、ステップ S306 で読み出した署名者識別子 sid と比較する (ステップ S312)。 rid と sid が等しくなければ、レコードが不連続である旨のエラーを返し、終了し、等しければ、ステップ S313 に進む。処理すべきレコードが物流情報に残っているか、確認し (ステップ S313)、残っていないければ検証に成功した旨を返し、終了し、残っていれば、ステップ S304 で読み出した被署名部 $data$ から受領者識別子を読み出し、変数 rid にセットし (ステップ S314)、次のレコードを処理するためにステップ S303 に戻る。

【0060】

本実施例ではステップ S303 からステップ S314 を繰り返すことにより、ステップ S303 で読み込む個々のレコード毎に署名の検証を行うが、個々のレコードの署名値 $s[1]$ 、 $s[2]$ 、 \dots 、 $s[m]$ (m はレコードの個数)、個々のレコードの被署名部に対するハッシュ値 (ステップ S307 の計算結果) $h[1]$ 、 $h[2]$ 、 \dots 、 $h[m]$ とステップ S302 で読み込む検証鍵 E 、 n を使い次式が成り立つかどうかを確認することで、署名の検証を行うようにしてもよい。

【0061】

【数 6】

$$(s[1] \cdot s[2] \cdot \dots \cdot s[m])^E \equiv h[1] \cdot h[2] \cdot \dots \cdot h[m] \pmod{n}$$

ステップ S 3 1 1 およびステップ S 3 1 2 は、ステップ S 3 0 6 の直後で行うようにしても良い。

【0062】

図4は、データキャリア1に記憶される物流情報に新しいレコードを追加する場合の物流情報処理モジュール2での処理の流れを示している。

【0063】

図4において、まず、読み取り手段4によってデータキャリア1に物流情報が格納されているかどうかを確認し（ステップ S 4 0 1）、物流情報が格納されていなければ、物流情報生成手段11は、新しいレコードの製品識別子 p i d にデータキャリアを添付する製品の識別子をセットし（ステップ S 4 0 2 a）、新しいレコードの署名者識別子 s i d に署名者の識別子をセットし（ステップ S 4 0 3 a）、ステップ S 4 0 7 へ進む。ステップ S 4 0 2 a、S 4 0 3 a で与える識別子はユーザがその都度指定したり、適当なリストから読み込んだり、あらかじめ埋め込まれているものを用いるなど、既知の手法によって与えられるものとし、その方法は問わない。

【0064】

ステップ S 4 0 1 で物流情報が格納されていた場合、読み取り手段4によってデータキャリアから物流情報を読み込み（ステップ S 4 0 2 b）、読み込んだ物流情報から最後のレコードを読み出す（ステップ S 4 0 3 b）。読み出したレコードから被署名部を読み出し（ステップ S 4 0 4 b）、その被署名部から製品識別子を読み出し、新しいレコードの製品識別子 p i d にセットする（ステップ S 4 0 5 b）。ステップ S 4 0 4 b で読み出した被署名部から受領者識別子を読み出し、新しいレコードの署名者識別子 s i d にセットする（ステップ S 4 0 6 b）。こののちステップ S 4 0 7 へ進む。

【0065】

さて、ステップ 4 0 7 において、新しいレコードの受領者識別子 r i d に製品の出荷先（受領者）の識別子をセットする。新しいレコードの他のフィールドに適切な値をセットする（ステップ S 4 0 8）。ステップ S 4 0 7、S 4 0 8 でセットする値は、ステップ S 4 0 2 a、S 4 0 3 a 同様、ユーザが指定する、適当

なりリストから読み込む、あらかじめ埋め込まれているなど既知の手法によって与えられ、特に指定の方法は限定しない。上記ステップによって作成された新しいレコードの被署名部を取出し、変数 $data$ にセットする（ステップ S409）。署名手段 16 は、署名者秘密情報記憶手段 17 から署名者識別子を取り出し、 sid' にセットする（ステップ S4010）。新しいレコードの署名者識別子 sid と、署名者秘密情報記憶手段 17 から読み出した署名者識別子 sid' とを比較し（ステップ S411）、ステップ S411 で sid と sid' が等しくなければ、署名者がおかしい旨のエラーを返し、終了し、等しければ、関数 $H(data, pid, sid)$ によってハッシュ値を計算し、変数 h にセットする（ステップ S412）。ここで使われる関数 $H(data, pid, sid)$ は、物流情報検証処理のハッシュ値の計算（図 3 のステップ S307）で用いられるものと同じである。第一の署名者秘密情報記憶手段 17 から署名者秘密情報を取出し変数 d にセットする（ステップ S413）。署名鍵情報選択手段 13 は、署名鍵情報記憶手段 15 から製品識別子 pid に対応する署名鍵情報を取出し、変数 t 、 n にセットする（ステップ S414）。署名手段 16 は、第一の署名値を次式によって計算し、変数 $r1$ にセットする（ステップ S415）。

【0066】

【数 7】

$$h^{f(d,n,pid,sid)} \bmod n$$

ここで、関数 $f()$ は引数 d 、 n 、 pid 、 sid から一意の値を返し、その値から d 、 n 、 pid 、 sid の値を知ることができない一方向性を持つ関数である。例えば、 $d|n|pid|sid$ （ $|$ はビット列の連結）に対して返す値に対して SHA-1、MD5 などハッシュ関数を適用したものであるが、これに限定されない。物流情報生成手段 11 は、第二の署名値を次式で計算する（ステップ S416）。

【0067】

【数 8】

$$h^t \bmod n$$

ステップ S415、ステップ S416 の計算結果 $r1$ と $r2$ を使って、次式によ

り署名値を計算し、変数 $sign$ にセットする（ステップ S417）。

【0068】

【数10】

$$(r1 \cdot r2) \bmod n$$

ステップ S417 で計算された署名値 $sign$ を新しいレコードの署名値にセットする（ステップ S418）。以上により物流情報に追加する新しいレコードが生成されたので、データキャリア 1 の物流情報に書き込み手段 5 により新しいレコードを追加する（ステップ S419）。

【0069】

本実施例では、署名鍵情報は既に署名鍵情報記憶手段 15 に記憶されているものとしたが、署名鍵情報は事前に署名情報取得処理により取得しても良いし、ステップ S413 で署名鍵情報が見つからない場合に、署名情報取得処理を実行することにより、取得しても良い。また、上記手順では、ステップ S415 は署名手段 16 で行い、ステップ S416 は物流情報生成手段 11 で行うが、ステップ S416 も署名手段 16 で行うようにしても良いし、ステップ S415 とステップ S416 は署名手段 16 と物流情報生成手段 11 とで平行して計算するようにしてもよい。

【0070】

図 5 は、署名鍵情報取得処置の流れを示している。署名鍵取得手段 15 は、データキャリア 1 から物流情報を読み出す（ステップ S501）。署名者秘密情報保持手段 17 から署名者識別子を読み出す（ステップ S502）。物流情報と署名者識別子を第一の通信手段 8 を通じて、物流情報管理モジュール 3 へ送る（ステップ S503）。物流情報管理モジュール 3 では、第二の通信手段で受け取った物流情報を第二の物流情報検証手段 19 に渡し、検証処理を行う（ステップ S504）。検証処理は図 3 で示した第一の物流情報検証処理と同じである。ステップ S504 で検証に失敗した場合は、物流情報が正しくない旨を第二の通信手段 18 を通じ、物流情報処理モジュール 2 へ送り、終了し、成功した場合は、署名者秘密情報選択手段 25 は第二の署名者秘密情報記憶手段 26 から第二の通信手段 18 で受け取った署名者識別子 sid に対応する署名者秘密情報を読み出し

、変数 d にセットする（ステップ S505、S506）。署名鍵情報生成手段 23 は署名鍵記憶手段から署名鍵を読み出し、変数 D 、 n にセットする（ステップ S507）。署名者識別子 sid とステップ S504 の検証処理中に読み出した製品識別子 pid およびステップ S507 で読み出した署名鍵を使い次式を計算し、変数 t にセットする（ステップ S508）。

【0071】

【数 11】

$$D = f(d, n, pid, sid)$$

ここで $f()$ は、前記署名生成処理のステップ S415 と同じ関数である。署名鍵情報として t 、 n を第二の通信手段 18 を通じて、物流情報処理モジュール 3 へ送る（ステップ S509）。第一の通信手段 18 を通じて、署名鍵情報を受け取ると、署名鍵情報取得手段 15 は署名鍵情報記憶手段 14 へ製品識別子と対応づけて格納する（ステップ S510）。ここで D は物流情報検証手段 6 で署名検証に使われる E に対応する秘密情報であり、 n は E と共に検証に使われる公開情報である。署名アルゴリズムに RSA 署名を用いる場合、

【0072】

【数 12】

$$a^{E \cdot D} \bmod n = a$$

なる関係を持つ値である。

【0073】

また、署名鍵情報生成モジュールで、受け付けた製品識別子と署名者識別子の組を記録しておき、署名鍵情報生成処理を開始する前に、既に同じ製品識別子と署名者識別子の組を受け付けたことがないかを確認し、新しい組み合わせに対してのみ署名鍵情報を生成する、あるいは、再発行であることを記録に残すようにしてもよい。またこの記録をある製品識別子に付いて調べることで、その製品が流通過程でどこにあるかを知ることができ、特定の署名者識別子に付いて調べることで、その署名者の在庫状況を知ることが可能となる。

【0074】

なお、物流情報のフォーマットとしては図 2 に示すものに限定されず、種々の

データ構造を採用できる。たとえば、図2に示すフォーマットにヘッダー情報を付加しても良い。すなわち、レコード1の前に検証鍵識別子を付与しても良い。図2のデータ構造を用いた処理においては検証鍵識別子は既知のものとして処理が行われているが、ヘッダー情報を付加することによりデータキャリア（タグ）ごとに検証鍵を付与することができる。この場合、図3のステップS302（検証鍵の読み出し）に代えて、ヘッダー情報に含まれる検証鍵識別子を用いて検証鍵を読み出すようにすればよい。

【0075】

〔実施例2〕

図6は、第一の物流情報検証部6の他の構成例を示している。第一の物流情報検証部6は、第一の情報検証手段9と、第一の検証鍵記憶手段10と、第一の検証鍵選択手段29とから構成されている。

【0076】

図7は、データキャリア1に記憶される物流情報の他の例を示している。データキャリア1には被署名部と署名値、必要ならその他の情報を一つのレコードとして物流情報が記憶されており、被署名部は少なくとも検証鍵識別子、製品識別子、署名者識別子、受領者識別子を含んでいる。

【0077】

図8は、データキャリア1から読み出した図7の物流情報を処理する場合の本実施例における第一の情報検証部6での物流情報検証処理の流れを示している。

【0078】

図8において、読み取り手段4によりデータキャリア1から読み出された物流情報が送られてくると、第一の情報検証手段9は、受領者識別子用の変数 *rid* をクリアする（ステップS801）。物流情報から最初のレコードを読み出す（ステップS802）。読み出したレコードから被署名部を読み出し、変数 *data* にセットする（ステップS803）。*data* から製品識別子を読み出し、変数 *pid* にセットする（ステップS804）。*data* から署名者識別子を読み出し、変数 *sid* にセットする（ステップS805）。関数 $H(data, pid, sid)$ によってハッシュ値を計算し、変数 *h* にセットする（ステップ

S806)。ここでハッシュ関数 $H()$ は引数 $data$ 、 pid 、 sid から一意の値を返す関数 $F(data, pid, sid)$ 、例えば、 $data | pid | sid$ ($|$ はビット列の連結)、の返す値に対してSHA-1、MD5などハッシュ関数を適用したものであり、関数 $F()$ 、ハッシュ関数は特定のものに限定されない。ステップS802で読み出したレコードから署名部を読み出し、変数 $sign$ にセットする(ステップS807)。ステップS803で読み出した被署名部 $data$ から検証鍵識別子を読み出し、変数 kid にセットする(ステップS808)。第一の署名検証鍵記憶手段10から検証鍵識別子 kid に対応する署名検証鍵を読み出し、変数 E 、変数 n にセットする(ステップS809)。検証用の値を次式によって計算し、変数 val にセットする(ステップS810)。

【0079】

【数13】

$$sign^E \bmod n \quad (\bmod \text{は剰余演算})$$

ここで、ステップS810は署名アルゴリズムとしてRSA署名を用いている。ステップS810で他の署名アルゴリズムを用いる場合は、第一の署名検証鍵記憶手段10に記憶され、ステップS809で読み込まれる鍵はそのアルゴリズムに適したものが使用される。ステップS806で計算したハッシュ値 h とステップS810で計算した値 val が等しいか確認する。等しくなければ、署名がおかしい旨のエラーを返し、終了し、等しければ、ステップS812に進む(ステップS811)。 rid に値がセットされているか確認し(ステップS812)、 rid に値がセットされていない場合は、ステップS814へ進み、 rid に値がセットされていれば、ステップS805で読み出した署名者識別子 sid と比較する。 rid と sid が等しくなければ、レコードが不連続である旨のエラーを返し、終了し、等しければ、ステップS814に進む(ステップS813)。処理すべきレコードが物流情報に残っているか、確認し(ステップS814)、残っていない場合は検証に成功した旨を返し、終了し、残っていれば、ステップS803で読み出した被署名部 $data$ から受領者識別子を読み出し、変数 rid にセットし(ステップS815)、次のレコードを処理するためにステップS8

02に戻る。

【0080】

ステップS812およびステップS813はステップS806の前に行うようにしても良い。

【0081】

[実施例3]

図9は、署名モジュール12の他の構成例を示している。署名モジュール12は、署名手段16と第一の署名者秘密情報記憶手段17と署名鍵使用限度保持手段27とから構成されている。

【0082】

図10は、本実施例における署名生成処理の流れを示している。図10において、読み取り手段4によってデータキャリア1に物流情報が格納されているかどうかを確認し（ステップS1001）、物流情報が格納されていない場合は、物流情報生成手段11は、新しいレコードの製品識別子pidにデータキャリアを添付する製品の識別子をセットし（ステップS1002a）、新しいレコードの署名者識別子sidに署名者の識別子をセットし（ステップS1003a）、ステップS1007へ進む。ステップS1002a、S1003aで与える識別子はユーザが都度指定する、適当なリストから読み込む、あらかじめ埋め込まれているなど既知の手法によって与えられるものとし、その方法は問わない。ステップS1001で物流情報が格納されていた場合、読み取り手段4によってデータキャリアから物流情報を読み込み（ステップS1002b）、読み込んだ物流情報から最後のレコードを読み出す（ステップS1003b）。読み出したレコードから被署名部を読み出し（ステップS1004b）、その被署名部から製品識別子を読み出し、新しいレコードの製品識別子pidにセットする（ステップS1005b）。ステップS1004bで読み出した被署名部から受領者識別子を読み出し、新しいレコードの署名者識別子sidにセットする（ステップS1006b）。

【0083】

ステップS1003aまたはステップS1006bに続いて、新しいレコード

の受領者識別子 $r i d$ に製品の出荷先（受領者）の識別子をセットする（ステップ $S 1 0 0 7$ ）。新しいレコードの他のフィールドに適切な値をセットする（ステップ $S 1 0 0 8$ ）。ステップ $S 1 0 0 7$ 、 $S 1 0 0 8$ でセットする値は、ステップ $S 1 0 0 2 a$ 、 $S 1 0 0 3 a$ 同様、ユーザが指定する、適当なリストから読み込む、あらかじめ埋め込まれているなど既知の手法によって与えられ、特に指定の方法は限定しない。上記ステップによって作成された新しいレコードの被署名部を取出し、変数 $d a t a$ にセットする（ステップ $S 1 0 0 9$ ）。署名手段 16 は、署名者秘密情報記憶手段 17 から署名者識別子を取出し、 $s i d'$ にセットする（ステップ $S 1 0 1 0$ ）。新しいレコードの署名者識別子 $s i d$ と、署名者秘密情報記憶手段 17 から読み出した署名者識別子 $s i d'$ とを比較し（ステップ $S 1 0 1 1$ ）、ステップ $S 1 0 1 1$ で $s i d$ と $s i d'$ が等しくなければ、署名者がおかしい旨のエラーを返し、終了し、等しければ、署名鍵使用限度情報保持手段 27 から製品識別子 $p i d$ に対応する署名鍵情報の署名鍵使用限度情報を読み出し、変数 m にセットする（ステップ $S 1 0 1 2$ ）。 m が正の値でなければ、該当する署名鍵は規定の使用回数に達している旨のエラーを返し、終了し、 m が正の値であれば、ステップ $S 1 0 1 4$ に進み（ステップ $S 1 0 1 3$ ）、 m から 1 減じた値を m にセットする（ステップ $S 1 0 1 4$ ）。署名鍵使用限度情報保持手段 27 に記憶されている署名鍵使用限度情報を変数 m の値に更新する（ステップ $S 1 0 1 5$ ）。ステップ $S 1 0 0 2 a$ またはステップ $S 1 0 0 5 b$ でセットした製品識別子 $p i d$ 、ステップ $S 1 0 0 3 a$ またはステップ $S 1 0 0 6 b$ でセットした署名者識別子 $s i d$ 、ステップ $S 1 0 0 9$ でセットした被署名部 $d a t a$ をもとにハッシュ値 $H(d a t a, p i d, s i d)$ を計算し、変数 h にセットする（ステップ $S 1 0 1 6$ ）。ここでハッシュ関数 $H()$ は引数 $d a t a$ 、 $p i d$ 、 $s i d$ から一意の値を返す関数 $F(d a t a, p i d, s i d)$ 、例えば、 $d a t a | p i d | s i d$ （ $|$ はビット列の連結）、の返す値に対して $S H A - 1$ 、 $M D 5$ などハッシュ関数を適用したものであり、関数 $F()$ 、ハッシュ関数は特定のものに限定されない。署名手段 16 は第一の署名者秘密情報記憶手段 17 から署名者秘密情報を読み出し、変数 d にセットする（ステップ $S 1 0 1 7$ ）。署名鍵情報選択手段 13 は製品識別子 $p i d$ に対応する署名鍵情報を署名鍵

情報記憶手段 14 から読み出し、変数 t 、 n にセットする（ステップ S1018）。署名手段 16 は、第一の署名値を次式により計算し、変数 r_1 にセットする（ステップ S1019）。

【0084】

【数 14】

$$h^{f(d,n,pid,sid)} \bmod n$$

ここで、関数 $f()$ は引数 d 、 n 、 pid 、 sid から一意の値を返し、その値から d 、 n 、 pid 、 sid の値を知ることができない一方向性を持つ関数である。例えば、 $d|n|pid|sid$ （ $|$ はビット列の連結）に対して返す値に対して SHA-1、MD5 などハッシュ関数を適用したものであるが、これに限定されない。物流情報生成手段 11 は、第二の署名値を次式で計算し、変数 r_2 にセットする（ステップ S1020）。

【0085】

【数 15】

$$h^t \bmod n$$

ステップ S1019、ステップ S1020 の計算結果 r_1 、 r_2 を使って次式により署名値を計算し、変数 $sign$ にセットする（ステップ S1021）。

【0086】

【数 16】

$$(r_1 \cdot r_2) \bmod n$$

ステップ S1021 で計算された署名値 $sign$ を新しいレコードの署名値にセットし（ステップ S1022）、上のステップで生成された新しいレコードを書き込み手段 5 によりデータキャリア 1 内の物流情報に追加する（ステップ S1023）。

【0087】

本実施例では署名鍵情報は既に署名鍵情報記憶手段 15 に記憶されているものとしたが、署名鍵情報は事前に署名情報取得処理により取得しても良いし、ステップ S1018 で署名鍵情報が見つからない場合に、署名情報取得処理を実行することにより、取得しても良い。また、上記手順では、ステップ S1019 は署

名手段 16 で、ステップ S1020 は物流情報生成手段 11 で行うが、ステップ S1020 も署名手段 16 で行うようにしても良いし、ステップ S1019 とステップ S1020 は署名手段 16 と物流情報生成手段 11 とで平行して計算するようにしてもよい。

【0088】

〔実施例 4〕

図 11 は、図 9 の署名モジュール 12 の処理の流れを示す他の実施例である。図 11 において、読み取り手段 4 によってデータキャリア 1 に物流情報が格納されているかどうかを確認し（ステップ S1101）、物流情報が格納されていない場合は、物流情報生成手段 11 は、新しいレコードの製品識別子 *p i d* にデータキャリアを添付する製品の識別子をセットし（ステップ S1102a）、新しいレコードの署名者識別子 *s i d* に署名者の識別子をセットし（ステップ S1103a）、ステップ S1107 へ進む。ステップ S1102a、S1103a で与える識別子はユーザが都度指定する、適当なリストから読み込む、あらかじめ埋め込まれているなど既知の手法によって与えられるものとし、その方法は問わない。ステップ S1101 で物流情報が格納されていた場合、読み取り手段 4 によってデータキャリアから物流情報を読み込み（ステップ S1102b）、読み込んだ物流情報から最後のレコードを読み出す（ステップ S1103b）。読み出したレコードから被署名部を読み出し（ステップ S1104b）、その被署名部から製品識別子を読み出し、新しいレコードの製品識別子 *p i d* にセットする（ステップ S1105b）。ステップ S1104b で読み出した被署名部から受領者識別子を読み出し、新しいレコードの署名者識別子 *s i d* にセットする（ステップ S1106b）。新しいレコードの受領者識別子 *r i d* に製品の出荷先（受領者）の識別子をセットする（ステップ S1107）。新しいレコードの他のフィールドに適切な値をセットする（ステップ S1108）。ステップ S1107、S1108 でセットする値は、ステップ S1102a、S1103a 同様、ユーザが指定する、適当なリストから読み込む、あらかじめ埋め込まれているなど既知の手法によって与えられ、特に指定の方法は限定しない。上記ステップによって作成された新しいレコードの被署名部を取出し、変数 *d a t a* にセットする

(ステップS1109)。署名手段16は、署名者秘密情報記憶手段17から署名者識別子を取り出し、sid' にセットする(ステップS1110)。新しいレコードの署名者識別子sidと、署名者秘密情報記憶手段17から読み出した署名者識別子sid' とを比較し(ステップS1111)、ステップS1111でsidとsid' が等しくなければ、署名者がおかしい旨のエラーを返し、終了し、等しければ、署名鍵使用限度情報保持手段27から製品識別子pidに対応する署名鍵情報の署名鍵使用限度情報の有無を確認する(ステップS1112)。署名鍵使用限度情報があれば、該当する署名鍵は使用済である旨のエラーを返し、終了し、署名鍵使用限度情報がなければ、ステップS1113に進み、製品識別子に対応されて署名鍵使用限度情報を署名鍵使用限度情報保持手段27に登録する(ステップS1113)。ステップS1102aまたはステップS1105bでセットした製品識別子pid、ステップS1103aまたはステップS1106bでセットした署名者識別子sid、ステップS1109でセットした被署名部dataをもとにハッシュ値H(data, pid, sid)を計算し、変数hにセットする(ステップS1114)。ここでハッシュ関数H()は引数data、pid、sidから一意の値を返す関数F(data, pid, sid)、例えば、data|pid|sid(|はビット列の連結)、の返す値に対してSHA-1、MD5などハッシュ関数を適用したものであり、関数F()、ハッシュ関数は特定のものに限定されない。署名手段16は第一の署名者秘密情報記憶手段17から署名者秘密情報を読み出し、変数dにセットする(ステップS1115)。署名鍵情報選択手段13は製品識別子pidに対応する署名鍵情報を署名鍵情報記憶手段14から読み出し、変数t、nにセットする(ステップS1116)。署名手段16は、第一の署名値を次式により計算し、変数r1にセットする(ステップS1117)。

【0089】

【数17】

$$h^{f(d,n,pid,sid)} \bmod n$$

ここで、関数f()は引数d、n、pid、sidから一意の値を返し、その値からd、n、pid、sidの値を知ることができない一方向性を持つ関数であ

る。例えば、 $d | n | p i d | s i d$ （ $|$ はビット列の連結）に対して返す値に対してSHA-1、MD5などハッシュ関数を適用したものであるが、これに限定されない。物流情報生成手段11は、第二の署名値を次式で計算し、変数 $r2$ にセットする（ステップS1118）。

【0090】

【数18】

$$h^t \bmod n$$

ステップS1117、ステップS1118の計算結果 $r1$ 、 $r2$ を使って次式により署名値を計算し、変数 $sign$ にセットする（ステップS1119）。

【0091】

【数19】

$$(r1 \cdot r2) \bmod n$$

ステップS1119で計算された署名値 $sign$ を新しいレコードの署名値にセットし（ステップS1120）、上のステップで生成された新しいレコードを書き込み手段5によりデータキャリア1内の物流情報に追加する（ステップS1121）。本実施例では署名鍵情報は既に署名鍵情報記憶手段15に記憶されているものとしたが、署名鍵情報は事前に署名情報取得処理により取得しても良いし、ステップS1116で署名鍵情報が見つからない場合に、署名情報取得処理を実行することにより、取得しても良い。また、上記手順では、ステップS1117は署名手段16で、ステップS1118は物流情報生成手段11で行うが、ステップS1118も署名手段16で行うようにしても良いし、ステップS1117とステップS1118は署名手段16と物流情報生成手段11とで平行して計算するようにしてもよい。

【0092】

【実施例5】

図12は、署名鍵情報生成部20の他の構成例を示している。署名鍵情報生成部20は、署名鍵情報生成手段23と署名者秘密情報選択手段25と第二の署名者秘密鍵情報記憶手段26と署名鍵記憶手段24と署名鍵選択手段31とから構成されている。

【0093】

図13は、署名鍵情報取得処理の流れを示している。図13において、署名鍵取得手段15は、データキャリア1から物流情報を読み出す（ステップS1301）。署名者秘密情報保持手段17から署名者識別子を読み出す（ステップS1302）。物流情報と署名者識別子を第一の通信手段8を通じて、物流情報管理モジュール3へ送る（ステップS1303）。物流情報管理モジュール3では、第二の通信手段18で受け取った物流情報を第二の物流情報検証手段19に渡し、検証処理を行う（ステップS1304）。検証処理は図8で示した第一の物流情報検証処理と同じである。ステップS1304で検証に失敗した場合は、物流情報が正しくない旨を第二の通信手段18を通じ、物流情報処理モジュール2へ送り、終了し、成功した場合は、署名者秘密情報選択手段25は第二の署名者秘密情報記憶手段26から第二の通信手段18で受け取った署名者識別子に対応する署名者秘密情報を読み出し、変数dにセットする（ステップS1305、S1306）。署名鍵情報生成手段23は、ステップS1304の検証処理中に取り出した検証鍵識別子kidに対応する署名鍵を署名鍵記憶手段から読み出し、変数D、nにセットする（ステップS1307）。署名者識別子sidとステップS1304の検証処理中に読み出した製品識別子pidおよびステップS1307で読み出した署名鍵を使い次式を計算し、変数tへセットする（ステップS1308）。

【0094】

【数20】

$$D = f(d, n, pid, sid)$$

ここでf（）は、前記署名生成処理（図10のステップS1019や図11のステップS1117）と同じ関数である。署名鍵情報としてt、n、kidを第二の通信手段18を通じて、物流情報処理モジュール3へ送る（ステップS1309）。第一の通信手段18を通じて、署名鍵情報を受け取ると、署名鍵情報取得手段15は署名鍵情報記憶手段14へ製品識別子と対応づけて格納する（ステップS1310）。ここでDは物流情報検証手段6で署名検証に使われるEに対応する秘密情報であり、nはEと共に検証に使われる公開情報である。署名アルゴ

リズムにRSA署名を用いる場合、

【0095】

【数21】

$$a^{E \cdot D} \bmod n = a$$

なる関係を持つ値となる。

【0096】

【実施例6】

図14は、データキャリア1に格納される物流情報の一実施例を示している。データキャリア1には被署名部と必要ならその他の情報を一つのレコードとして、複数のレコードからなるデータ部と、署名値からなる物流情報が記憶されており、被署名部は少なくとも製品識別子、署名者識別子、受領者識別子を含んでいる。

【0097】

図15は、図14に示される物流情報を検証するために物流情報検証部6で行われる検証処理の流れを示している。図15において、データキャリア1から読み取られた物流情報が与えられると、変数valを初期化し（1をセット。ステップS1501）、第一の検証鍵記憶手段10から署名検証鍵を読み出し、変数E、nにセットする（ステップS1502）。物流情報からデータ部を読み出す（ステップS1503）。読み出したデータ部から1レコード読み出す（ステップS1504）、読み出したレコードから被署名部を取出し、変数dataにセットする（ステップS1505）。被署名部から製品識別子を読み出し、変数pidにセットし（ステップS1506）、被署名部から署名者識別子を取出し、変数sidにセットする（ステップS1508）。ステップS1505で読み出した被署名部data、ステップS1506で読み出した製品識別子pid、ステップS1507で読み出した署名者識別子sidを使いハッシュ値H(data, pid, sid)を計算し、変数hにセットする（ステップS1508）。ここでハッシュ関数H()は引数data、pid、sidから一意の値を返す関数F(data, pid, sid)、例えば、data|pid|sid(|はビット列の連結)、の返す値に対してSHA-1、MD5などハッシュ関数を

適用したものであり、関数 $F()$ 、ハッシュ関数は特定のものに限定されない。
次式を計算し、変数 val にセットする（ステップ S1509）。

【0098】

【数22】

$$(val \cdot h) \bmod n$$

ステップ S1503 で読み出したデータ部に未処理のレコードがあるか確認する（ステップ S1510）。未処理のレコードがあればステップ S1504 へ戻り、次のレコードを処理する。未処理のレコードがなければ、物流情報から署名値を読み出し、変数 $sign$ にセットする（ステップ S1511）。ステップ S1502 で読み出した検証鍵 E 、 n とステップ S1511 で読み出した署名値 $sign$ を使って、次式により検証値を計算し、変数 $val2$ にセットする（ステップ S1512）。

【0099】

【数23】

$$sign^E \bmod n$$

変数 val と $val2$ の値を比較し（ステップ S1513）、等しければ検証に成功した旨を返し、終了し、等しくなければ検証に失敗した旨を返し、終了する。

【0100】

図16は、図14に示される物流情報を生成するために物流情報生成部6における処理の流れを示している。データキャリア1に物流情報があるか確認し（ステップ S1601）、物流情報が格納されていなければ、物流情報生成手段11は、新しいレコードの製品識別子 pid にデータキャリアを添付する製品の識別子をセットし（ステップ S1602a）、新しいレコードの署名者識別子 sid に署名者の識別子をセットし（ステップ S1603a）、ステップ S1608に進む。ステップ S1602a、S1603a で与える識別子はユーザが都度指定する、適当なリストから読み込む、あらかじめ埋め込まれているなど既知の手法によって与えられるものとし、その方法は問わない。ステップ S1601 で物流情報が格納されていた場合、読み取り手段4によってデータキャリアから物流情

報を読み込み（ステップS1602b）、読み込んだ物流情報からデータ部を読み出す（ステップS1603b）。読み込んだデータ部から最後のレコードを読み出し（ステップS1604b）、読み出したレコードから被署名部を読み出す（ステップS1605b）。その被署名部から製品識別子を読み出し、新しいレコードの製品識別子 pid にセットする（ステップS1606b）。ステップS1604bで読み出した被署名部から受領者識別子を読み出し、新しいレコードの署名者識別子 sid にセットする（ステップS1607b）。新しいレコードの受領者識別子 rid に製品の出荷先（受領者）の識別子をセットする（ステップS1608）。新しいレコードの他のフィールドに適切な値をセットする（ステップS1609）。新しいレコードから被署名部を取出し、変数 $data$ へセットする（ステップS1610）。署名手段16は、署名者秘密情報記憶手段17から署名者識別子を取出し、 sid' にセットする（ステップS1611）。新しいレコードの署名者識別子 sid と、署名者秘密情報記憶手段17から読み出した署名者識別子 sid' とを比較し（ステップS1612）、ステップS1612で sid と sid' が等しくなければ、署名者がおかしい旨のエラーを返し、終了し、等しければ、ステップS1602aまたはステップS1606bでセットした製品識別子 pid 、ステップS1603aまたはステップS1607bでセットした署名者識別子 sid 、ステップS1610でセットした被署名部 $data$ をもとにハッシュ値 $H(data, pid, sid)$ を計算し、変数 h にセットする（ステップS1613）。ここでハッシュ関数 $H()$ は、物流情報検証部6で行われる検証処理（図15のステップS1508）で使われるハッシュ関数と同じである。署名手段16は第一の署名者秘密情報記憶手段17から署名者秘密情報を読み出し、変数 d にセットする（ステップS1614）。署名鍵情報選択手段13は製品識別子 pid に対応する署名鍵情報を署名鍵情報記憶手段14から読み出し、変数 t 、 n にセットする（ステップS1615）。署名手段16は、第一の署名値を次式により計算し、変数 $r1$ にセットする（ステップS1616）。

【0101】

【数24】

$$h^{f(d,n,pid,sid)} \bmod n$$

ここで、関数 $f()$ は引数 d 、 n 、 pid 、 sid から一意の値を返し、その値から d 、 n 、 pid 、 sid の値を知ることができない一方向性を持つ関数である。例えば、 $d \parallel n \parallel pid \parallel sid$ (\parallel はビット列の連結) に対して返す値に対して SHA-1、MD5 などハッシュ関数を適用したものであるが、これに限定されない。物流情報生成手段 11 は、第二の署名値を次式で計算し、変数 r_2 にセットする (ステップ S1617)。

【0102】

【数25】

$$h^t \bmod n$$

ステップ S1616、ステップ S1617 の計算結果 r_1 、 r_2 を使って次式により署名値を計算し、変数 s にセットする (ステップ S1618)。

【0103】

【数26】

$$(r_1 \cdot r_2) \bmod n$$

ステップ S1602b で読み出した物流情報から署名部を読み出し、変数 $sign$ にセットする (ステップ S1619)。ステップ S1615 で読み出した n 、ステップ S1618 で計算した s 、ステップ S1619 で読み出した $sign$ を使い、次式を計算し、変数 $sign$ にセットする (ステップ S1620)。

【0104】

【数27】

$$(sign \cdot s) \bmod n$$

データキャリア 1 のデータ部に上記ステップで生成された新しいレコードを追加し (ステップ S1621)、ステップ S1620 で計算された変数 $sign$ の値でデータキャリア 1 の署名値を更新する (ステップ S1622)。

【0105】

本実施例では署名鍵情報は既に署名鍵情報記憶手段 15 に記憶されているものとしたが、署名鍵情報は事前に署名情報取得処理により取得しても良いし、ステップ S1615 で署名鍵情報が見つからない場合に、署名情報取得処理を実行す

ることにより、取得しても良い。また、上記手順では、ステップ S1616 は署名手段 16 で、ステップ S1617 は物流情報生成手段 11 で行うが、ステップ S1617 も署名手段 16 で行うようにしても良いし、ステップ S1616 とステップ S1617 は署名手段 16 と物流情報生成手段 11 とで平行して計算するようにしてもよい。

【0106】

〔実施例 7〕

図 17 は、図 14 に示される物流情報を検証するために物流情報検証部 6 で行われる検証処理の流れを示している。図 17 において、読み取り手段 4 によりデータキャリア 1 から物流情報を読み取り（ステップ S1701）、読み取った物流情報からデータ部を読み出し（ステップ S1702）、変数 *rid*, *data* をクリアする（ステップ S1703）。ステップ S1702 で読み出したデータ部から 1 レコード読み出し（ステップ S1704）、読み出したレコードから被署名部を読み出し、変数 *d* にセットする（ステップ S1705）。ステップ S1705 で読み出した被署名部から署名者識別子を読み出し、変数 *sid* にセットする（ステップ S1706）。変数 *rid* に値がセットされているか確認し（ステップ S1707）、セットされていないならば、ステップ S1709 に進み、セットされていれば、ステップ S1706 で読み出した署名者識別子 *sid* と *rid* を比較する（ステップ S1708）。*sid* と *rid* が等しくなければ、レコードが不連続である旨を返し、終了し、等しければステップ S1709 に進む。ステップ S1705 で読み出した被署名部 *d* と変数 *data* の値を使って次式を計算し、変数 *data* にセットする（ステップ S1709）。

【0107】

【数 28】

$data \mid d$ （ \mid はビット列の連結。ただし *data* がクリアされている時は、*d* を返す）

ここでステップ S1709 での計算は他の方法でもよく、*data* と *d* を使った計算結果が *data* と *d* の組に対して一対一に対応していればよい。ステップ S1702 で読み出したデータ部に未処理のレコードがあるか確認し（ステップ S

1710)、未処理のレコードがあれば、ステップS1705で読み出した被署名部から受領者識別子を変数 rid にセットし(ステップS1711)、ステップS1704に戻り、次の未処理レコードの処理をおこない、未処理のレコードが無ければ、ステップS1712に進む。ステップS1705で読み出した被署名部 d から製品識別子を読み出し、変数 pid にセットする(ステップS1712)。上記ステップにより計算された変数 $data$ 、ステップS1706で読み出した署名者識別子 sid 、ステップS1712で読み出した製品識別子 pid を使いハッシュ値 $H(data, pid, sid)$ を計算し、変数 h にセットする(ステップS1713)。ここでハッシュ関数 $H()$ は引数 $data$ 、 pid 、 sid から一意の値を返す関数 $F(data, pid, sid)$ 、例えば、 $data || pid || sid$ ($|$ はビット列の連結)、の返す値に対してSHA-1、MD5などハッシュ関数を適用したものであり、関数 $F()$ 、ハッシュ関数は特定のものに限定されない。ステップS1701で読み出した物流情報から署名値を読み出し、変数 $sign$ にセットする(ステップS1714)。第一の検証鍵記憶手段10から署名検証鍵を読み出し、変数 E, n にセットする(ステップS1715)。ステップS1714で読み出した署名値 $sign$ とステップS1715で読み出した署名検証鍵 E, n を使い次式により検証用の値を計算し、変数 val にセットする(ステップS1716)。

【0108】

【数29】

$$sign^E \bmod n$$

ステップS1713で計算したハッシュ値 h とステップS1716で計算した検証用の値 val とを比較する(ステップS1717)。 h と val が等しければ検証に成功した旨を返し、終了し、等しくなければ検証に失敗した旨を返し、終了する。

【0109】

図18は、図14に示される物流情報を生成するために物流情報生成部7で行われる処理の流れを示している。図18において、変数 $data$ を初期化し(ステップS1801)、データキャリア1内に物流情報があるか確認し(ステップ

S1802)、無ければ、物流情報のデータ部に記載するための新しいレコードの製品識別子 $p i d$ をセットし (ステップ S1803 a)、新しいレコードの署名者識別子 $s i d$ をセットし (ステップ 1804 a)、ステップ S1813 へ進む。ステップ S1802 で物流情報があれば、データキャリア 1 から読み取り手段 4 を介し物流情報を読み出し (ステップ S1803 b)、読み出した物流情報からデータ部を取出し (ステップ S1804 b)、そのデータ部から 1 レコード取出す (ステップ S1805 b)。取出したレコードの被署名部を取出し、変数 d にセットする (ステップ S1806 b)。変数 $d a t a$ とステップ S1806 b で取出した被署名部 d とから次式の計算を行い、変数 $d a t a$ にセットする (ステップ S1807 b)。

【0110】

【数30】

$d a t a | d$ (| はビット列の連結。ただし $d a t a$ がクリアされている時は、 d を返す)

ステップ S1807 b の計算は、他の方式でも良いが、物流情報検証部 6 で行われる検証処理での計算 (図 17 のステップ S1709) と同じであればよい。ステップ S1804 b で読み出したデータ部に未処理のレコードがあるか確認し (ステップ S1808 b)、未処理のレコードが無ければステップ S1809 b に進み、未処理のレコードがあれば、ステップ S1805 b に戻り、次のレコードを処理する。上記ステップにより読み出された被署名部 d から製品識別子を読み出し、新しいレコードの製品識別子 $p i d$ にセットし (ステップ S1809 b)、被署名部 b から受領者識別子 $r i d$ を読み出す (ステップ S1810 b)。署名者秘密情報保持手段 17 から署名者識別子を読み出し、新しいレコードの署名者識別子 $s i d$ にセットする (ステップ S1811 b)。ステップ 10 b で読み出した受領者識別子 $r i d$ とステップ S1811 b で読み出した署名者識別子 $s i d$ を比較し (ステップ S1812 b)、受領者識別子 $r i d$ と署名者識別子 $s i d$ が等しければ、ステップ S1813 へ進み、等しくなければ受領者と署名者が一致しない旨を返し、終了する。製品の受領者の識別子を新しいレコードの受領者識別子 $r i d$ にセットする (ステップ S1813)。新しいレコードの残り

のフィールドをセットし（ステップS1814）、新しいレコードの被署名部を取り出し、変数dへセットする（ステップS1815）。変数dataと変数dに対し、ステップS1807bと同じ計算を行い、変数dataにセットする（ステップS1816）。ステップS1803aまたはステップS1809bでセットした製品識別子pidと、ステップS1804aまたはステップS1811bでセットした署名者識別子sidと、ステップS1816でセットした変数dataに対し、ハッシュ値H(data, pid, sid)を計算する（ステップS1817）。ここでハッシュ関数H()は、物流情報検証部6で行われる検証処理（図17のステップS1713）で用いられるものと同じである。署名者秘密情報記憶手段17から署名者秘密情報を読み出し、変数dにセットし（ステップS1818）、署名鍵情報記憶手段14から製品識別子pidに対応する署名鍵情報を取出し、変数t, nにセットする（ステップS1819）。次式により、第一の署名値を計算し、変数s1にセットする（ステップS1820）。

【0111】

【数31】

$$h^{f(d,n,pid,sid)} \bmod n$$

ここで、関数f()は引数d、n、pid、sidから一意の値を返し、その値からd、n、pid、sidの値を知ることができない一方向性を持つ関数である。例えば、d|n|pid|sid(|はビット列の連結)に対して返す値に対してSHA-1、MD5などハッシュ関数を適用したものであるが、これに限定されない。次式により、第二の署名値を計算し、変数s2にセットする（ステップS1821）。

【0112】

【数32】

$$h^t \bmod n$$

ステップS1820で計算された第一の署名値s1、ステップS1821で計算された第二の署名値s2を使い次式により署名値を計算し、変数signにセットする（ステップS1822）。

【0113】

【数 33】

$$s1 \cdot s2 \bmod n$$

書き込み手段 5 を介し、データキャリア 1 内のデータ部に新しいレコードを追加し（ステップ S1823）、データキャリア 1 内の署名値をステップ S1822 で計算した変数 $sign$ の値で更新する（ステップ S1824）。

【0114】

〔実施例 8〕

図 19 は、データキャリア 1 に格納される物流情報の一実施例を示している。データキャリア 1 には被署名部と必要ならその他の情報を一つのレコードとして、複数のレコードからなるデータ部と、検証鍵識別子と署名値を一つのレコードとして、複数のレコードからなる署名値部とからなる物流情報が記憶されており、被署名部は少なくとも検証鍵識別子、製品識別子、署名者識別子、受領者識別子を含んでいる。

【0115】

図 20 は、図 19 に示される物流情報を検証するために物流情報検証部 6 で行われる検証処理の流れを示している。図 20 において、データキャリアからデータ部を読み出す（ステップ S2001）、読み出したデータ部から 1 レコード読み出す（ステップ S2002）。読み出したレコードから、被署名部を取り出し、変数 $data$ にセットする（ステップ S2003）。被署名部から、製品識別子を読み出し、変数 pid にセットする（ステップ S2004）。被署名部から、受領者識別子を読み出し、変数 sid にセットする（ステップ S2005）。ステップ S2003 で読み出した被署名部 $data$ 、ステップ S2004 で読み出した製品識別子 pid 、ステップ S2005 で読み出した署名者識別子 sid を使いハッシュ値 $H(data, pid, sid)$ を計算し、変数 h にセットする（ステップ S2006）。被署名部から検証鍵識別子を読み出し、変数 kid にセットする（ステップ S2007）。検証鍵識別子 kid に対応するバッファ変数 $val[kid]$ が存在するかを確認し、存在しない場合は領域を確保する（ステップ S2008、S2009）。変数 $val[kid]$ に 1 をセットする（ステップ S2010）。署名検証鍵記憶手段 24 から kid に対応する署名検

証鍵法数を読み出し、変数 n にセットする（ステップ S2011）。次式を計算し、変数 $val[kid]$ にセットする（ステップ S2012）。

【0116】

【数34】

$$(val[kid] \cdot h) \bmod n$$

ステップ S2001 で読み出したデータ部に未処理のレコードがあるか確認する（ステップ S2013）。未処理のレコードがあればステップ S2002 へ戻り、次のレコードを処理する。未処理のレコードがなければ、物流情報から署名値部を読み出す（ステップ S2014）。読み出した署名値部から 1 レコード読み出す（ステップ 15）。被署名部から検証鍵識別子を読み出し、変数 kid にセットする（ステップ S2016）。署名検証鍵記憶手段 24 から検証鍵識別子 kid に対応する署名検証鍵を読み出し、変数 E 、 n にセットする（ステップ S2017）。ステップ S2015 で読み出したレコードから署名値を読み出し、変数 $sign$ にセットする（ステップ S2018）。ステップ S2017 で読み出した検証鍵 E 、 n とステップ S2018 で読み出した署名値 $sign$ を使って、次式により検証値を計算し、変数 $val2$ にセットする（ステップ S2019）。

【0117】

【数35】

$$sign^E \bmod n$$

変数 $val[kid]$ と $val2$ の値を比較し（ステップ S2020）、等しくなければ検証に失敗した旨を返し、終了する。等しければ、ステップ S2014 で読み出した署名値部に未処理のレコードがあるか確認する（ステップ S2021）。未処理のレコードがあればステップ S2015 へ戻り、次のレコードを処理する。未処理のレコードがなければ、検証に成功した旨を返し、終了する。

【0118】

図 21 は、図 19 に示される物流情報を生成するために物流情報生成部 6 における処理の流れを示している。図 21 において、データキャリア 1 に物流情報があるか確認し（ステップ S2101）、物流情報が格納されていなければ、物流

情報生成手段 11 は、新しいレコードの製品識別子 $p i d$ にデータキャリアを添付する製品の識別子をセットし（ステップ $S 2102 a$ ）、新しいレコードの署名者識別子 $s i d$ に署名者の識別子をセットし（ステップ $S 2103 a$ ）、ステップ $S 2110$ に進む。ステップ $S 2102 a$ 、 $S 2103 a$ で与える識別子はユーザが都度指定する、適当なリストから読み込む、あらかじめ埋め込まれているなど既知の手法によって与えられるものとし、その方法は問わない。ステップ $S 2101$ で物流情報が格納されていた場合、読み取り手段 4 によってデータキャリアから物流情報を読み込み（ステップ $S 2102 b$ ）、読み込んだ物流情報からデータ部を読み出す（ステップ $S 2103 b$ ）。読み込んだデータ部から最後のレコードを読み出し（ステップ $S 2104 b$ ）、読み出したレコードから被署名部を読み出す（ステップ $S 2105 b$ ）。その被署名部から製品識別子を読み出し、新しいレコードの製品識別子 $p i d$ にセットする（ステップ $S 2106 b$ ）。ステップ $S 2104 b$ で読み出した被署名部から受領者識別子を読み出し、新しいレコードの署名者識別子 $s i d$ にセットする（ステップ $S 2107 b$ ）。署名手段 16 は、署名者秘密情報記憶手段 17 から署名者識別子を取り出し、 $s i d'$ にセットする（ステップ $S 2108 b$ ）。新しいレコードの署名者識別子 $s i d$ と、署名者秘密情報記憶手段 17 から読み出した署名者識別子 $s i d'$ とを比較し（ステップ $S 2109 b$ ）、ステップ $S 2109 b$ で $s i d$ と $s i d'$ が等しくなければ、署名者がおかしい旨のエラーを返し、終了する。新しいレコードの受領者識別子 $r i d$ に製品の出荷先（受領者）の識別子をセットする（ステップ $S 2110$ ）。署名鍵記憶手段 14 から署名鍵情報を読み出し、署名鍵情報を t へ、署名鍵法数を n へ、署名鍵識別子を $i d$ へセットする（ステップ $S 2111$ ）。新しいレコードの検証鍵識別子 $k i d$ に署名鍵識別子 $i d$ をセットする（ステップ $S 2112$ ）。新しいレコードの他のフィールドに適切な値をセットする（ステップ $S 2113$ ）。新しいレコードから被署名部を取出し、変数 $d a t a$ へセットする（ステップ $S 2114$ ）。ステップ $S 2102 a$ またはステップ $S 2106 b$ でセットした製品識別子 $p i d$ 、ステップ $S 2103 a$ またはステップ $S 2107 b$ でセットした署名者識別子 $s i d$ 、ステップ $S 2114$ でセットした被署名部 $d a t a$ をもとにハッシュ値 $H(d a t a, p i d, s i d$

)を計算し、変数hにセットする(ステップS2115)。署名手段16は第一の署名者秘密情報記憶手段17から署名者秘密情報を読み出し、変数dにセットする(ステップS2116)。署名手段16は、第一の署名値を次式により計算し、変数r1にセットする(ステップS2117)。

【0119】

【数36】

$$h^{f(d,n,pid,sid)} \bmod n$$

ここで、関数f()は引数d、n、pid、sidから一意の値を返し、その値からd、n、pid、sidの値を知ることができない一方向性を持つ関数である。例えば、d|n|pid|sid(|はビット列の連結)に対して返す値に対してSHA-1、MD5などハッシュ関数を適用したものであるが、これに限定されない。物流情報生成手段11は、第二の署名値r2を次式で計算する(ステップS2118)。

【0120】

【数37】

$$h^t \bmod n$$

ステップS2117、ステップS2118の計算結果r1、r2を使って次式により署名値を計算し、変数sにセットする(ステップS2119)。

【0121】

【数38】

$$(r1 \cdot r2) \bmod n$$

データキャリア1から署名値部を読み出す(ステップS2120)。検証鍵識別子kidに対応する署名値があるか確認し、ある場合は、データキャリア1から署名値を読み出し、変数signにセットし(ステップS2121、S2122a)、ステップS2119でセットしたsと署名鍵法数nから、次式によりsignを計算する(ステップS2123a)。

【0122】

【数39】

$$(sign \cdot s) \bmod n$$

検証鍵識別子 *k i d* に対応する署名値が無い場合は、データキャリア 1 の署名値部に新しいレコードを追加し（ステップ S 2 1 2 2 b）、ステップ S 2 1 1 9 でセットした *s* を *s i g n* にセットする（ステップ S 2 1 2 3 b）。データキャリア 1 のデータ部に上記ステップで生成された新しいレコードを追加し（ステップ S 2 1 2 4）、ステップ S 2 1 2 3 a またはステップ S 2 1 2 3 b で計算された変数 *s i g n* の値でデータキャリア 1 の署名値部の検証鍵識別子 *k i d* に対応する署名値を更新する（ステップ S 2 1 2 5）。

【0123】

本実施例では署名鍵情報は既に署名鍵情報記憶手段 1 5 に記憶されているものとしたが、署名鍵情報は事前に署名情報取得処理により取得しても良いし、ステップ S 2 1 1 1 で署名鍵情報が見つからない場合に、署名情報取得処理を実行することにより、取得しても良い。また、上記手順では、ステップ S 2 1 1 7 は署名手段 1 6 で、ステップ S 2 1 1 8 は物流情報生成手段 1 1 で行うが、ステップ S 2 1 1 8 も署名手段 1 6 で行うようにしても良いし、ステップ S 2 1 1 7 とステップ S 2 1 1 8 は署名手段 1 6 と物流情報生成手段 1 1 とで平行して計算するようにしてもよい。

【0124】

[実施例 9]

図 2 2 は、図 1 9 に示される物流情報を検証するために物流情報検証部 6 で行われる検証処理の流れを示している。図 2 2 において、読み取り手段 4 によりデータキャリア 1 から物流情報を読み取り（ステップ S 2 2 0 1）、読み取った物流情報からデータ部を読み出し（ステップ S 2 2 0 2）、変数 *r i d*, *d a t a* をクリアする（ステップ S 2 2 0 3）。ステップ S 2 2 0 2 で読み出したデータ部から 1 レコード読み出し（ステップ S 2 2 0 4）、読み出したレコードから被署名部を読み出し、変数 *d* にセットする（ステップ S 2 2 0 5）。ステップ S 2 2 0 5 で読み出した被署名部から署名者識別子を読み出し、変数 *s i d* にセットする（ステップ S 2 2 0 6）。変数 *r i d* に値がセットされているか確認し（ステップ S 2 2 0 7）、セットされていなければ、ステップ S 2 2 0 9 に進み、セットされていれば、ステップ S 2 2 0 6 で読み出した署名者識別子 *s i d* と *r i*

dを比較する（ステップS2208）。sidとridが等しくなければ、レコードが不連続である旨を返し、終了し、等しければステップS2209に進む。ステップS2205で読み出した被署名部dと変数dataの値を使って次式を計算し、変数dataにセットする（ステップS2209）。

【0125】

【数40】

$data \mid d$ （|はビット列の連結。ただしdataがクリアされている時は、dを返す）。

ここでステップS2209での計算は他の方法でもよく、dataとdを使った計算結果がdataとdの組に対して一対一に対応していればよい。ステップS2202で読み出したデータ部に未処理のレコードがあるか確認し（ステップS2210）、未処理のレコードがあれば、ステップS2205で読み出した被署名部から受領者識別子を変数ridにセットし（ステップS2211）、ステップS2204に戻り、次の未処理レコードの処理をおこない、未処理のレコードが無ければ、ステップS2212に進む。ステップS2205で読み出した被署名部dから製品識別子を読み出し、変数pidにセットする（ステップS2212）。上記ステップにより計算された変数data、ステップS2206で読み出した署名者識別子sid、ステップS2212で読み出した製品識別子pidを使いハッシュ値 $H(data, pid, sid)$ を計算し、変数hにセットする（ステップS2213）。ここでハッシュ関数 $H()$ は引数data、pid、sidから一意の値を返す関数 $F(data, pid, sid)$ 、例えば、 $data \mid pid \mid sid$ （|はビット列の連結）、の返す値に対してSHA-1、MD5などハッシュ関数を適用したものであり、関数 $F()$ 、ハッシュ関数は特定のものに限定されない。ステップS2201で読み出した物流情報から署名値を読み出し、変数signにセットする（ステップS2214）。ステップS2205で読み出した被署名部dから検証鍵識別子を読み出し、変数kidにセットする（ステップS2215）。一の検証鍵記憶手段10からステップS2215で読み出した第検証鍵識別子kidに対応する署名検証鍵を読み出し、変数E、nにセットする（ステップS2216）。ステップS2214で読み出した

署名値 $sign$ とステップ S 2 2 1 6 で読み出した署名検証鍵 E 、 n を使い次式により検証用の値を計算し、変数 val にセットする（ステップ S 2 2 1 7）。

【0126】

【数41】

$$sign^E \bmod n$$

ステップ S 2 2 1 3 で計算したハッシュ値 h とステップ S 2 2 1 7 で計算した検証用の値 val とを比較する（ステップ S 2 2 1 8）。 h と val が等しければ検証に成功した旨を返し、終了し、等しくなければ検証に失敗した旨を返し、終了する。

【0127】

図 2 3 は、図 1 9 に示される物流情報を生成するために物流情報生成部 7 で行われる処理の流れを示している。図 2 3 において、変数 $data$ を初期化し（ステップ S 2 3 0 1）、データキャリア 1 内に物流情報があるか確認し（ステップ S 2 3 0 2）、無ければ、物流情報のデータ部に記載するための新しいレコードの製品識別子 pid をセットし（ステップ S 2 3 0 3 a）、新しいレコードの署名者識別子 sid をセットし（ステップ S 2 3 0 4 a）、ステップ S 2 3 1 3 へ進む。ステップ S 2 3 0 2 で物流情報があれば、データキャリア 1 から読み取り手段 4 を介し物流情報を読み出し（ステップ S 2 3 0 3 b）、読み出した物流情報からデータ部を取出し（ステップ S 2 3 0 4 b）、そのデータ部から 1 レコード取出す（ステップ S 2 3 0 5 b）。取出したレコードの被署名部を取出し、変数 d にセットする（ステップ S 2 3 0 6 b）。変数 $data$ とステップ S 2 3 0 6 b で取出した被署名部 d とから次式の計算を行い、変数 $data$ にセットする（ステップ S 2 3 0 7 b）。

【0128】

【数42】

$data \parallel d$ （ \parallel はビット列の連結。ただし $data$ がクリアされている時は、 d を返す）

ステップ S 2 3 0 7 b の計算は、他の方式でも良いが、物流情報検証部 6 で行われる検証処理での計算（図 2 2 のステップ S 2 2 0 9）と同じであればよい。ス

ステップ S 2 3 0 4 b で読み出したデータ部に未処理のレコードがあるか確認し（ステップ S 2 3 0 8 b）、未処理のレコードが無ければステップ S 2 3 0 9 b に進み、未処理のレコードがあれば、ステップ S 2 3 0 5 b に戻り、次のレコードを処理する。上記ステップにより読み出された被署名部 d から製品識別子を読み出し、新しいレコードの製品識別子 p i d にセットし（ステップ S 2 3 0 9 b）、被署名部 b から受領者識別子 r i d を読み出す（ステップ S 2 3 1 0 b）。署名者秘密情報保持手段 17 から署名者識別子を読み出し、新しいレコードの署名者識別子 s i d にセットする（ステップ S 2 3 1 1 b）。ステップ S 2 3 1 0 b で読み出した受領者識別子 r i d とステップ S 2 3 1 1 b で読み出した署名者識別子 s i d を比較し（ステップ S 2 3 1 2 b）、受領者識別子 r i d と署名者識別子 s i d が等しければ、ステップ S 2 3 1 3 へ進み、等しくなければ受領者と署名者が一致しない旨を返し、終了する。製品の受領者の識別子を新しいレコードの受領者識別子 r i d にセットする（ステップ S 2 3 1 3）。被署名部 d から署名検証鍵識別子を読み出し、新しいレコードの署名検証鍵識別子 k i d にセットする（ステップ S 2 3 1 4）。新しいレコードの残りのフィールドをセットし（ステップ S 2 3 1 5）、新しいレコードの被署名部を取り出し、変数 d へセットする（ステップ S 2 3 1 6）。変数 d a t a と変数 d に対し、ステップ S 2 3 0 7 b と同じ計算を行い、変数 d a t a にセットする（ステップ S 2 3 1 7）。ステップ S 2 3 0 3 a またはステップ S 2 3 0 9 b でセットした製品識別子 p i d と、ステップ S 2 3 0 4 a またはステップ S 2 3 1 1 b でセットした署名者識別子 s i d と、ステップ S 2 3 1 7 でセットした変数 d a t a に対し、ハッシュ値 $H(d a t a, p i d, s i d)$ を計算する（ステップ S 2 3 1 8）。ここでハッシュ関数 $H()$ は、物流情報検証部 6 で行われる検証処理（図 22 のステップ S 2 3 1 3）で用いられるものと同じである。署名者秘密情報記憶手段 17 から署名者秘密情報を読み出し、変数 d にセットし（ステップ S 2 3 1 9）、署名鍵情報記憶手段 14 から製品識別子 p i d に対応する署名鍵情報を取り出し、変数 t, n にセットする（ステップ S 2 3 2 0）。次式により、第一の署名値を計算し、変数 s 1 にセットする（ステップ S 2 3 2 1）。

【0129】

【数 4 3】

$$h^{f(d,n,pid,sid)} \bmod n$$

ここで、関数 $f()$ は引数 d 、 n 、 pid 、 sid から一意の値を返し、その値から d 、 n 、 pid 、 sid の値を知ることができない一方向性を持つ関数である。例えば、 $d | n | pid | sid$ ($|$ はビット列の連結) に対して返す値に対して SHA-1、MD5 などハッシュ関数を適用したものであるが、これに限定されない。次式により、第二の署名値を計算し、変数 s_2 にセットする (ステップ S2322)。

【0130】

【数 4 4】

$$h^t \bmod n$$

ステップ S2320 で計算された第一の署名値 s_1 、ステップ S2322 で計算された第二の署名値 s_2 を使い次式により署名値を計算し、変数 $sign$ にセットする (ステップ S2323)。

【0131】

【数 4 5】

$$(s_1 \cdot s_2) \bmod n$$

書き込み手段 5 を介し、データキャリア 1 内のデータ部に新しいレコードを追加し (ステップ S2324)、データキャリア 1 内の署名値をステップ S2323 で計算した変数 $sign$ の値で更新する (ステップ S2325)。

【0132】

【実施例 10】

図 24 は、図 1 の物流情報処理モジュール 2 および物流情報管理モジュール 3 をそれぞれ部分モジュールに分離した実施例を示している。図 24 において、物流情報処理モジュール 2 (図 1) を第一の検証モジュール 32 および書き込みモジュール 37 に分離している。また、物流情報管理モジュール 3 (図 1) を第二の検証モジュール 33 および署名鍵情報生成モジュール 34 に分離している。第一の検証モジュール 32 および書き込みモジュール 37 にはそれぞれ第 1 の読み取り手段 35 および第 2 の読み取り手段 36 が設けられている。また署名鍵情報

生成モジュール 3 4 には第三の通信手段 2 8 が設けられ、第二の検証モジュール 3 3 と通信を行うようになっている。なお、図 2 4 において図 1 または図 6 と対応する個所には対応する符号を付して説明を繰り返さない。

【0 1 3 3】

【発明の効果】

以上で説明したように、本発明によれば、流通業者が認証局から証明書を取得する必要が無く、また秘密情報が IC カードなどの耐タンパな容器に収められるためセキュリティ確保が容易であるので、多大な設備が必要なく管理が容易な流通管理システムを実現することができる。また、本発明において、署名回数を制限するようにすれば、偽造商品に本物の商品と同一の ID を振り、署名をして別の受け取り業者へ配送するといった不正を防ぐことができる。

【図面の簡単な説明】

【図 1】 本発明の実施例 1 を用いて本発明の基本の構成を説明するブロック図である。

【図 2】 実施例 1 の物流情報のデータ構造を説明する図である。

【図 3】 実施例 1 の検証時の流れを説明するフローチャートである。

【図 4】 実施例 1 の署名時の流れを説明するフローチャートである。

【図 5】 実施例 1 の署名鍵情報取得時の流れを説明するフローチャートである。

【図 6】 本発明の実施例 2 の情報検証部の構成を説明するブロック図である。

【図 7】 実施例 2 の物流情報のデータ構造を説明する図である。

【図 8】 実施例 2 の検証時の流れを説明するフローチャートである。

【図 9】 本発明の実施例 3 の署名モジュールの構成を説明するブロック図である。

【図 1 0】 実施例 3 の署名時の流れを説明するフローチャートである。

【図 1 1】 実施例 4 における署名モジュールの署名時の流れを説明するフローチャートである。

【図 1 2】 本発明の実施例 5 の署名鍵情報生成モジュールの構成を説明す

るブロック図である。

【図 13】 実施例 5 の署名鍵情報取得時の流れを説明するフローチャートである。

【図 14】 本発明の実施例 6 の物流情報のデータ構造を説明する図である。

【図 15】 実施例 6 の検証時の流れを説明するフローチャートである。

【図 16】 実施例 6 の署名時の流れを説明するフローチャートである。

【図 17】 実施例 7 の物流情報検証部における検証時の流れを説明するフローチャートである。

【図 18】 実施例 7 の署名時の流れを説明するフローチャートである。

【図 19】 本発明の実施例 8 の物流情報のデータ構造を説明する図である。

【図 20】 実施例 8 の検証時の流れを説明するフローチャートである。

【図 21】 実施例 8 の署名時の流れを説明するフローチャートである。

【図 22】 本発明の実施例 9 の検証時の流れを説明するフローチャートである。

【図 23】 実施例 9 の署名時の流れを説明するフローチャートである。

【図 24】 本発明の実施例 10 の構成を説明するブロック図である。

【符号の説明】

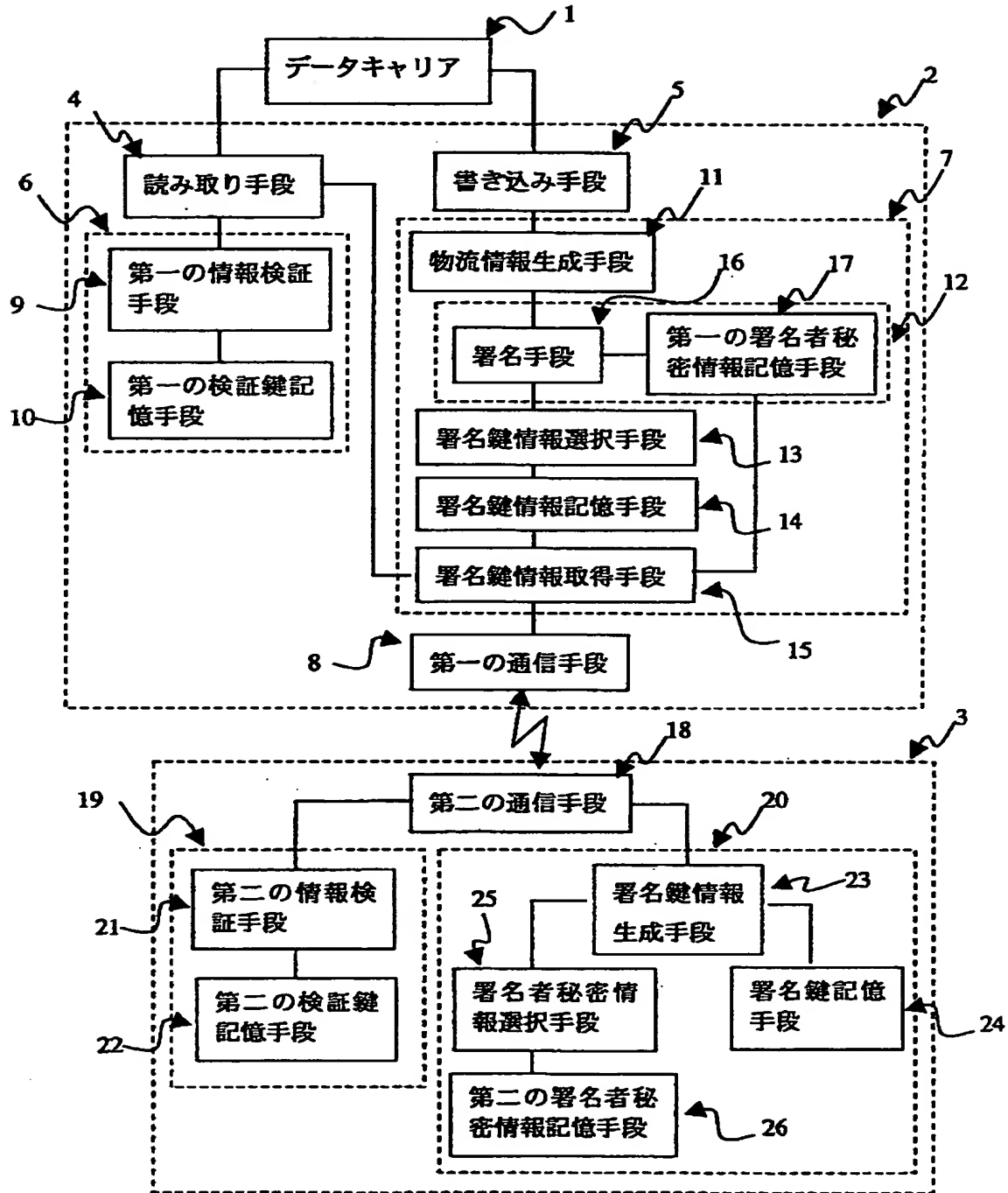
- 1 データキャリア
- 2 物流情報処理モジュール
- 3 物流情報管理モジュール
- 4 読み出し手段
- 5 書き込み手段
- 6 第一の情報検証部
- 7 情報生成部
- 8 第一の通信手段
- 9 第一の情報検証手段
- 10 第一の検証鍵記憶手段

- 1 1 物流情報生成手段
- 1 2 署名モジュール
- 1 3 署名鍵情報選択手段
- 1 4 署名鍵情報記憶手段
- 1 5 署名鍵情報取得手段
- 1 6 署名手段
- 1 7 第一の署名者秘密情報記憶手段
- 1 8 第二の通信手段
- 1 9 第二の情報検証部
- 2 0 署名鍵情報生成部
- 2 1 第二の情報検証手段
- 2 2 第二の検証鍵記憶手段
- 2 3 署名鍵情報生成手段
- 2 4 署名鍵記憶手段
- 2 5 署名者秘密情報選択手段
- 2 6 第二の署名者秘密情報記憶手段
- 2 7 署名鍵使用限度情報保持手段
- 2 8 第三の通信手段
- 2 9 第一の検証鍵選択手段
- 3 0 第二の検証鍵選択手段
- 3 1 署名鍵選択手段
- 3 2 第一の検証モジュール
- 3 3 第二の検証モジュール
- 3 4 署名鍵情報情報生成モジュール
- 3 5 第一の読み出し手段
- 3 6 第二の読み出し手段

【書類名】

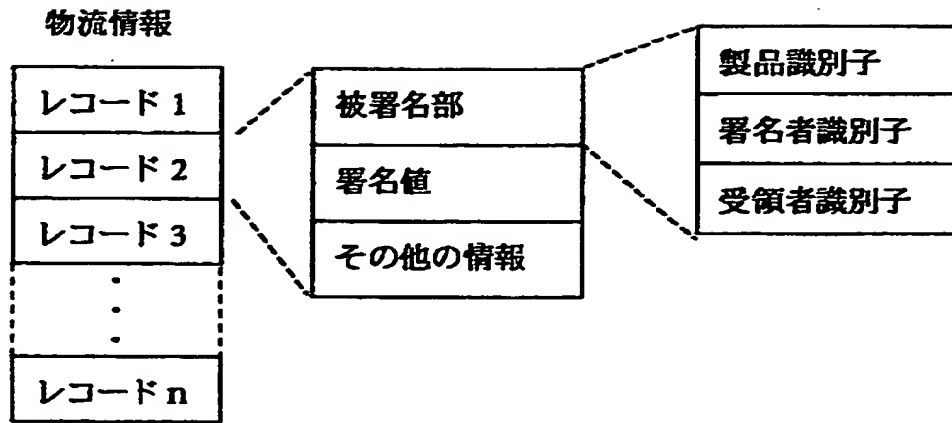
図面

【図 1】



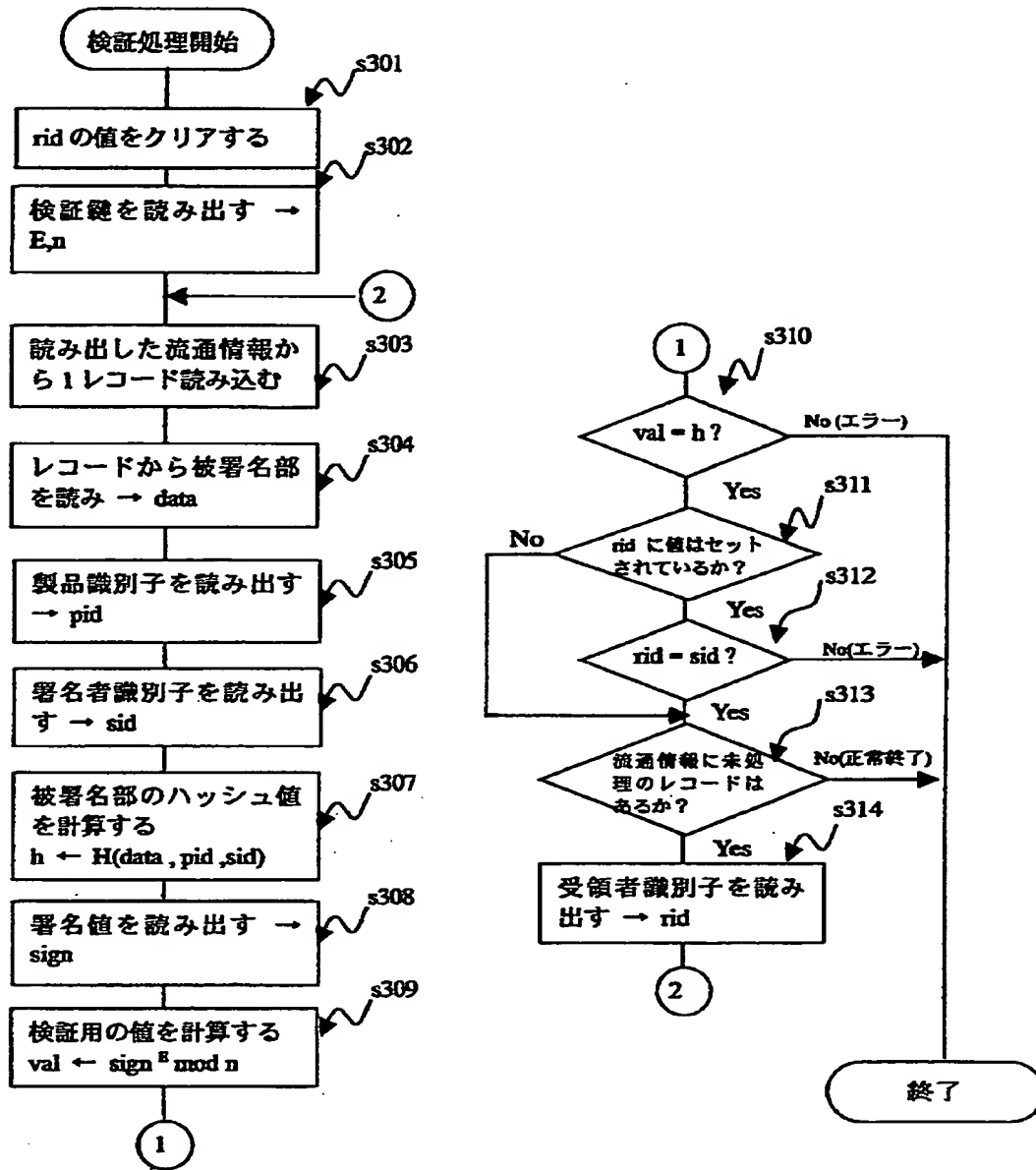
基本の構成

【図 2】

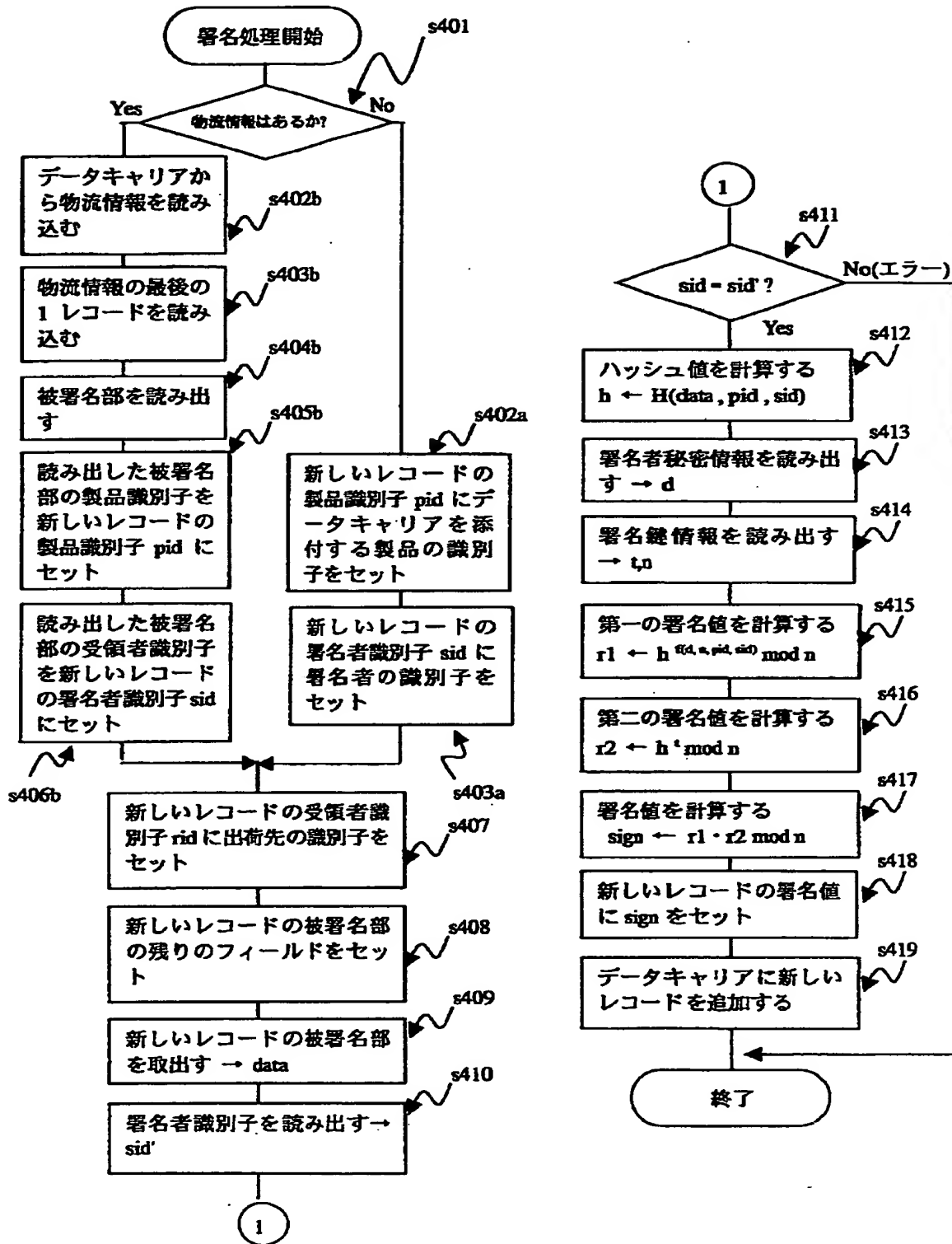


データキャリアに記憶される物流情報

【図 3】

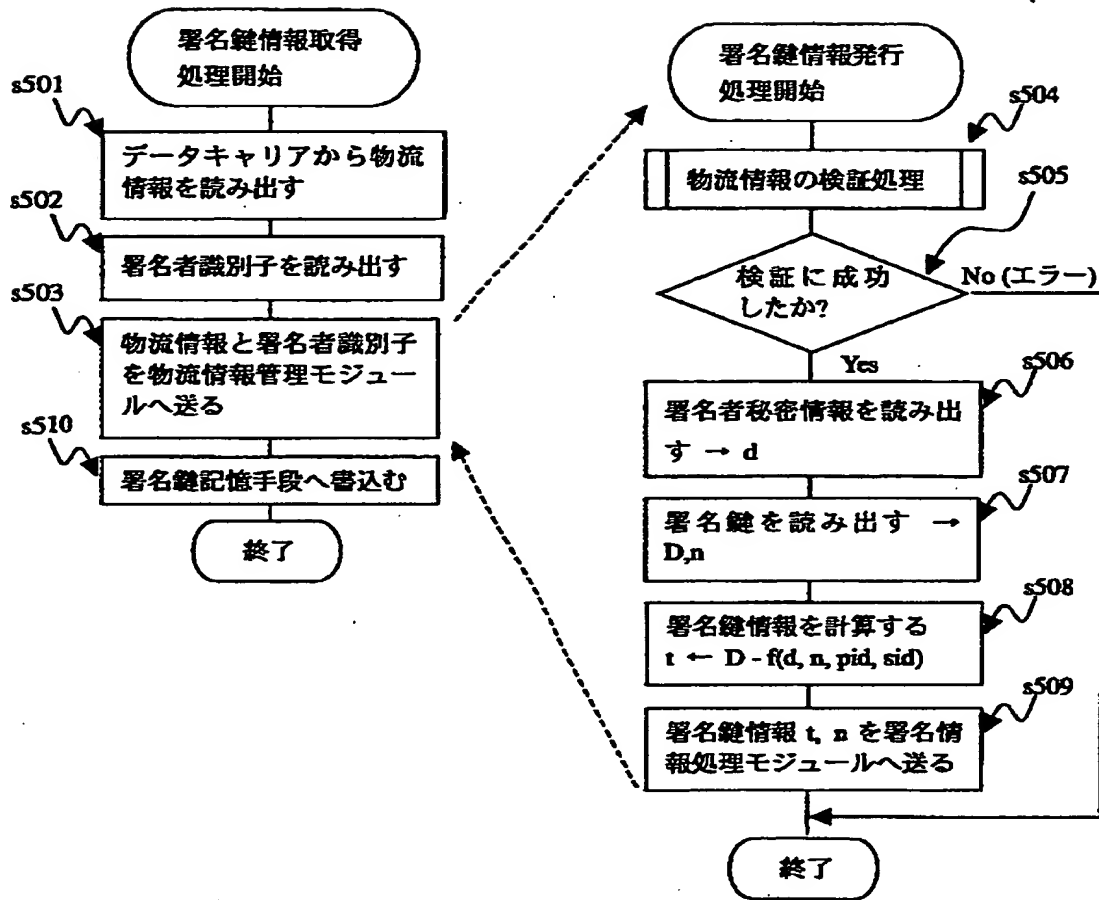


【図 4】



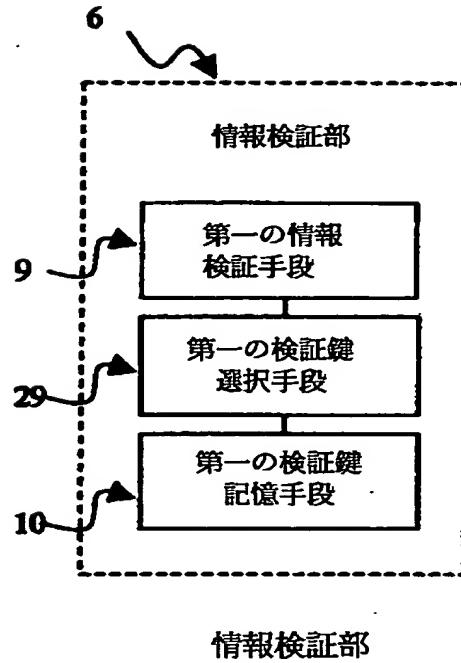
署名時の処理

【図 5】

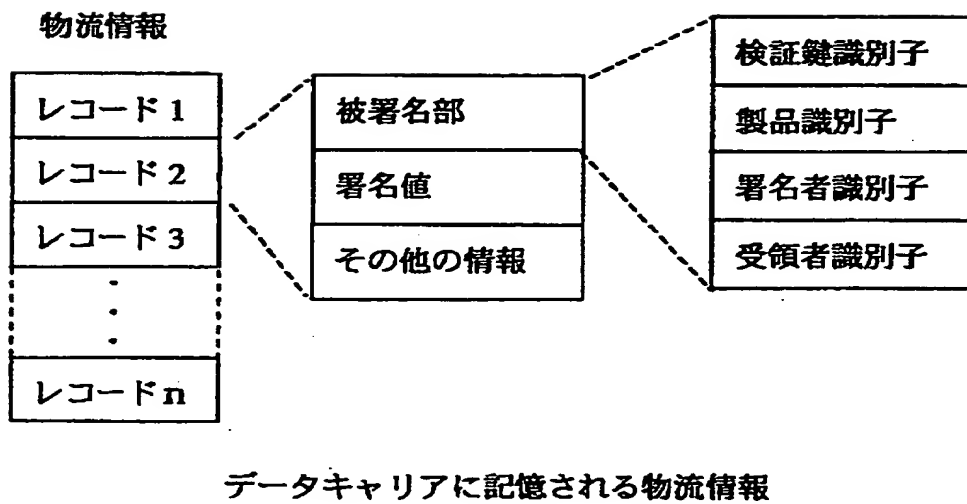


署名鍵情報取得時の処理

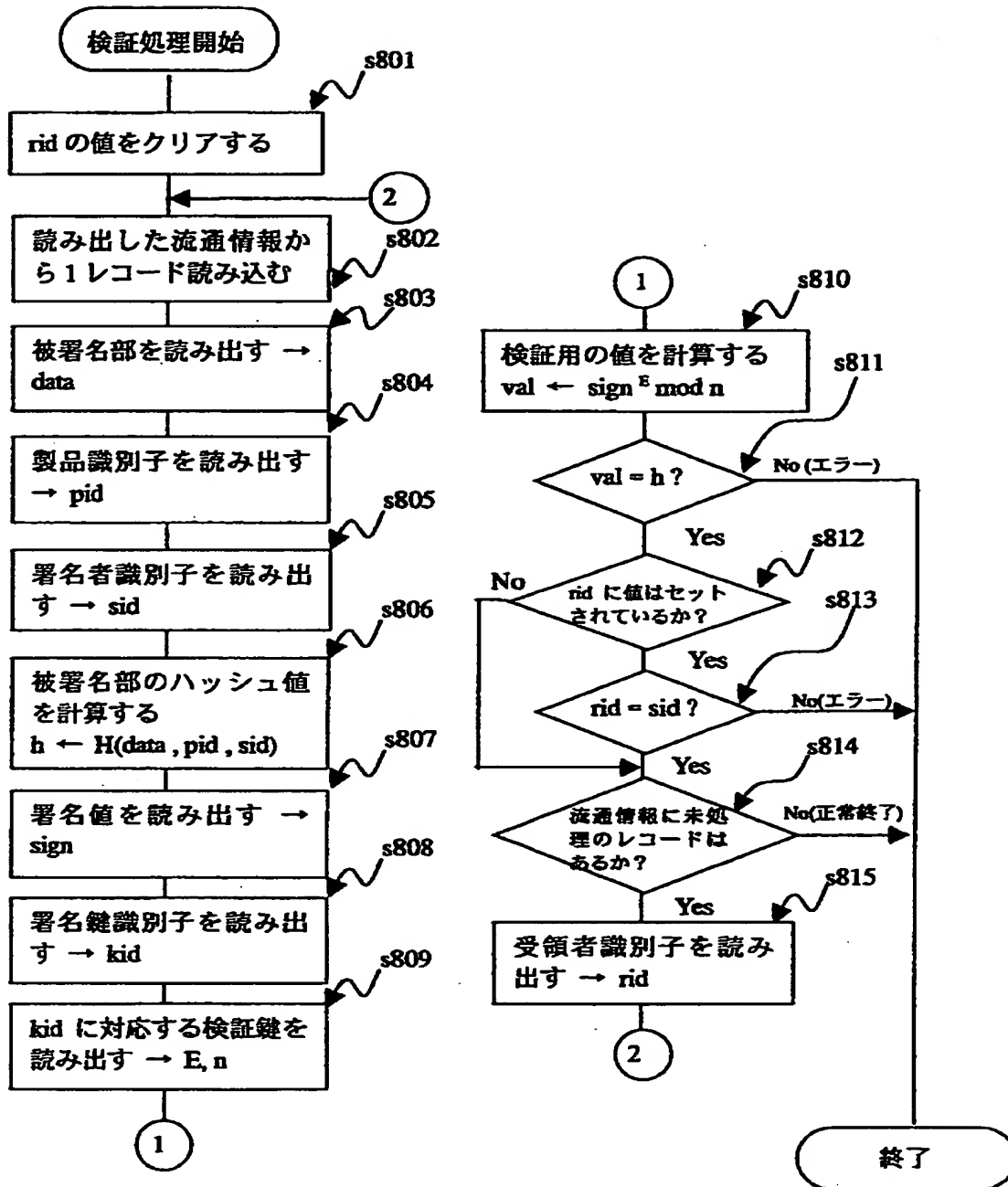
【図 6】



【図 7】

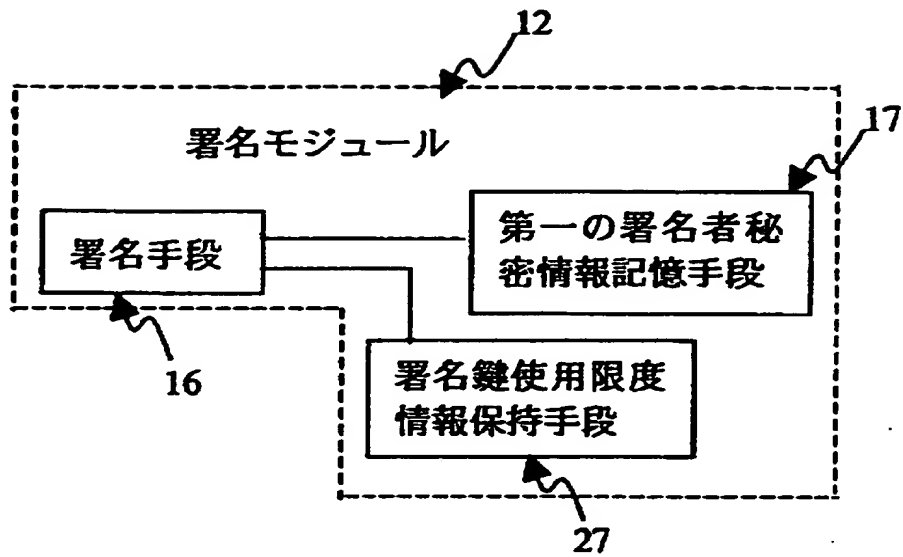


【図 8】



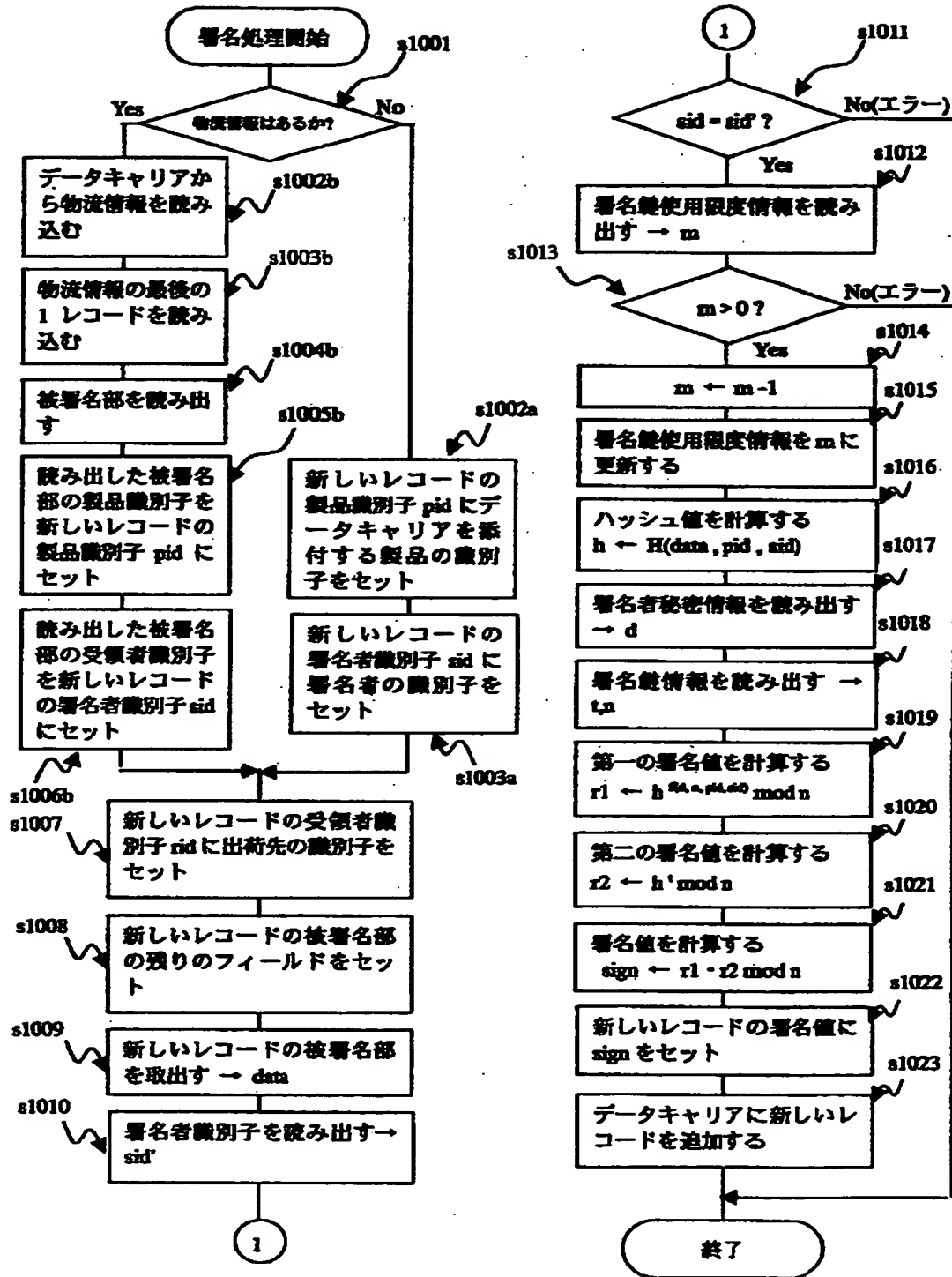
検証時の処理

【図9】



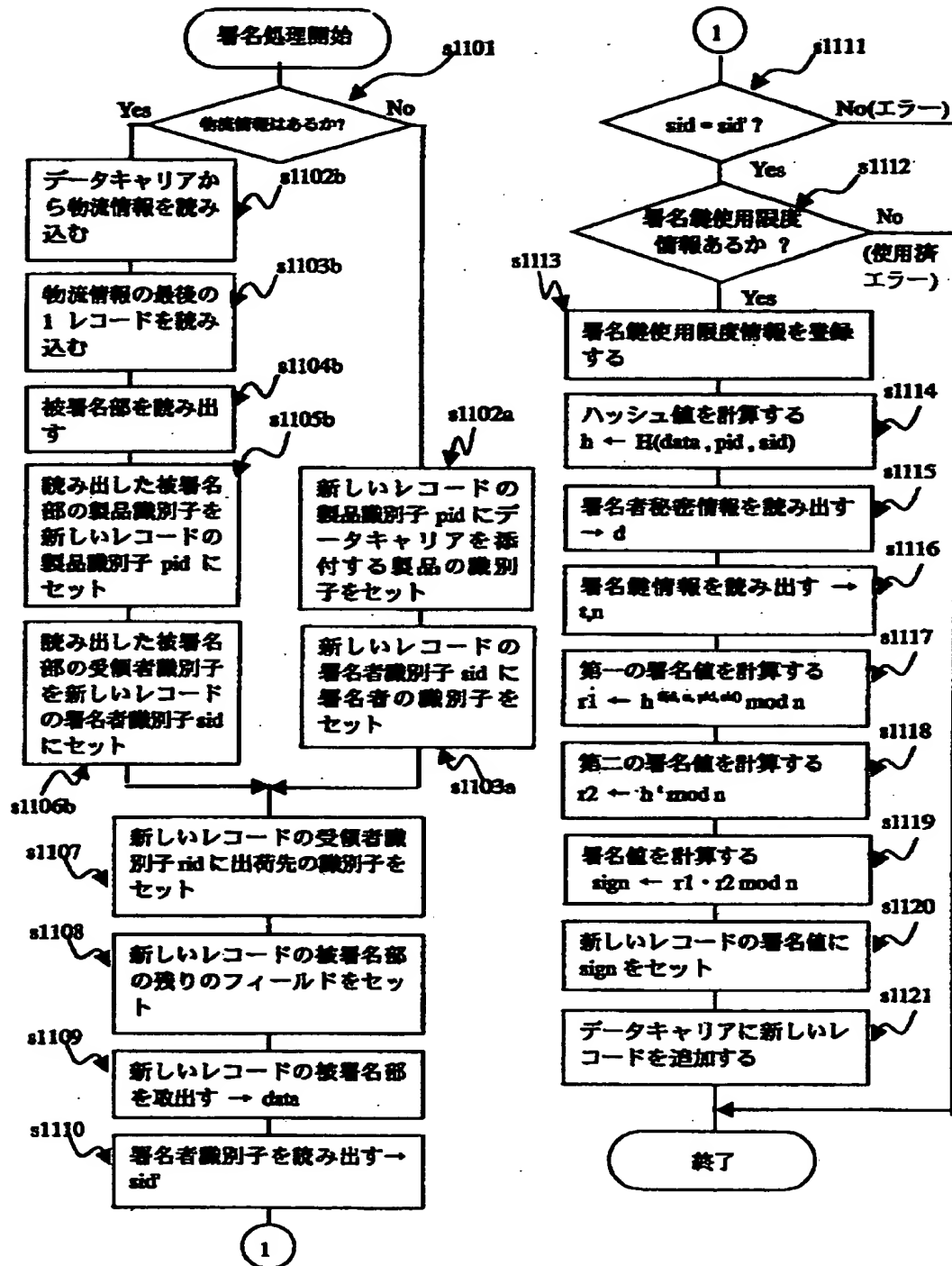
署名モジュール

【図 10】



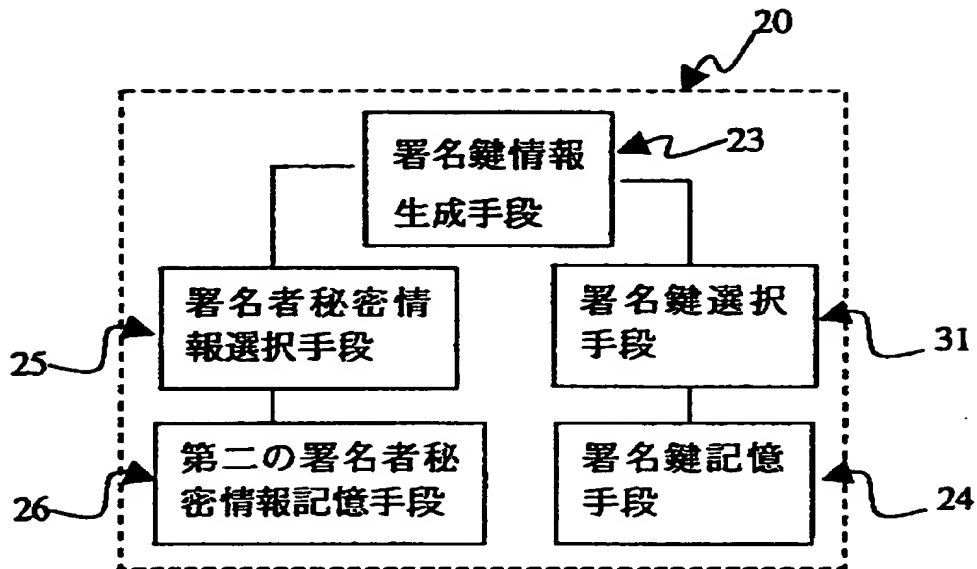
署名時の処理

【図 1 1】



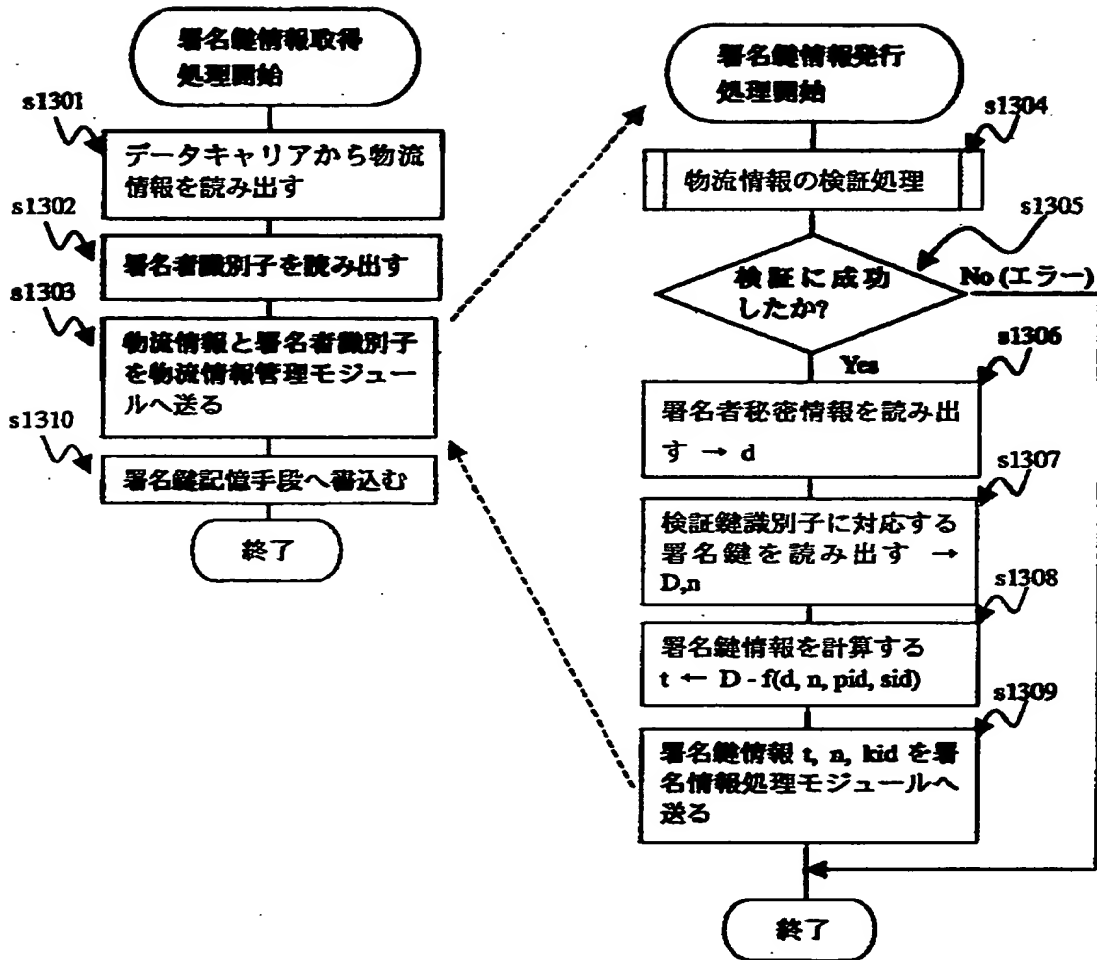
署名時の処理

【図 1 2】



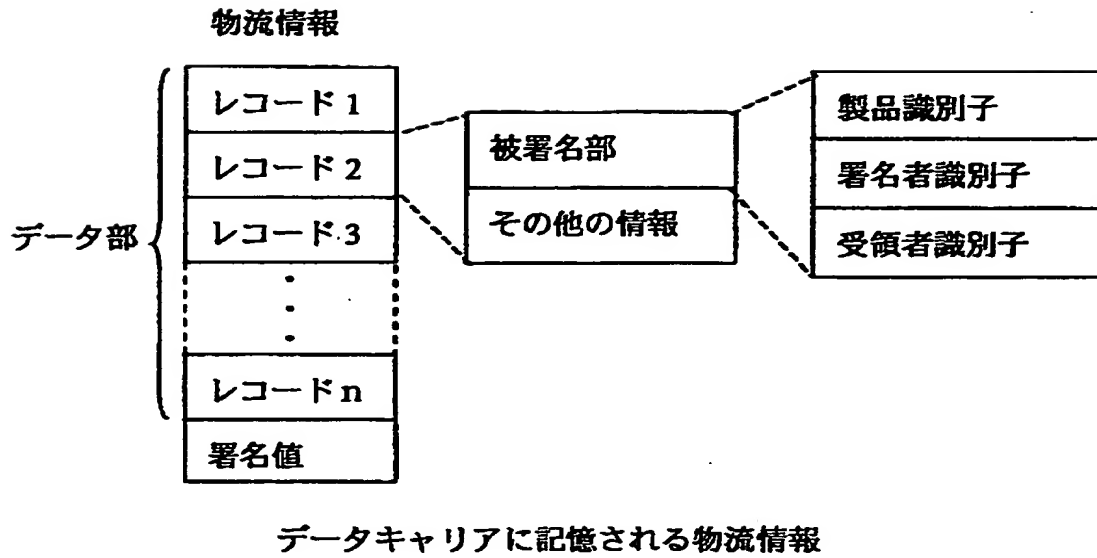
署名鍵情報情報生成モジュール

【図 13】

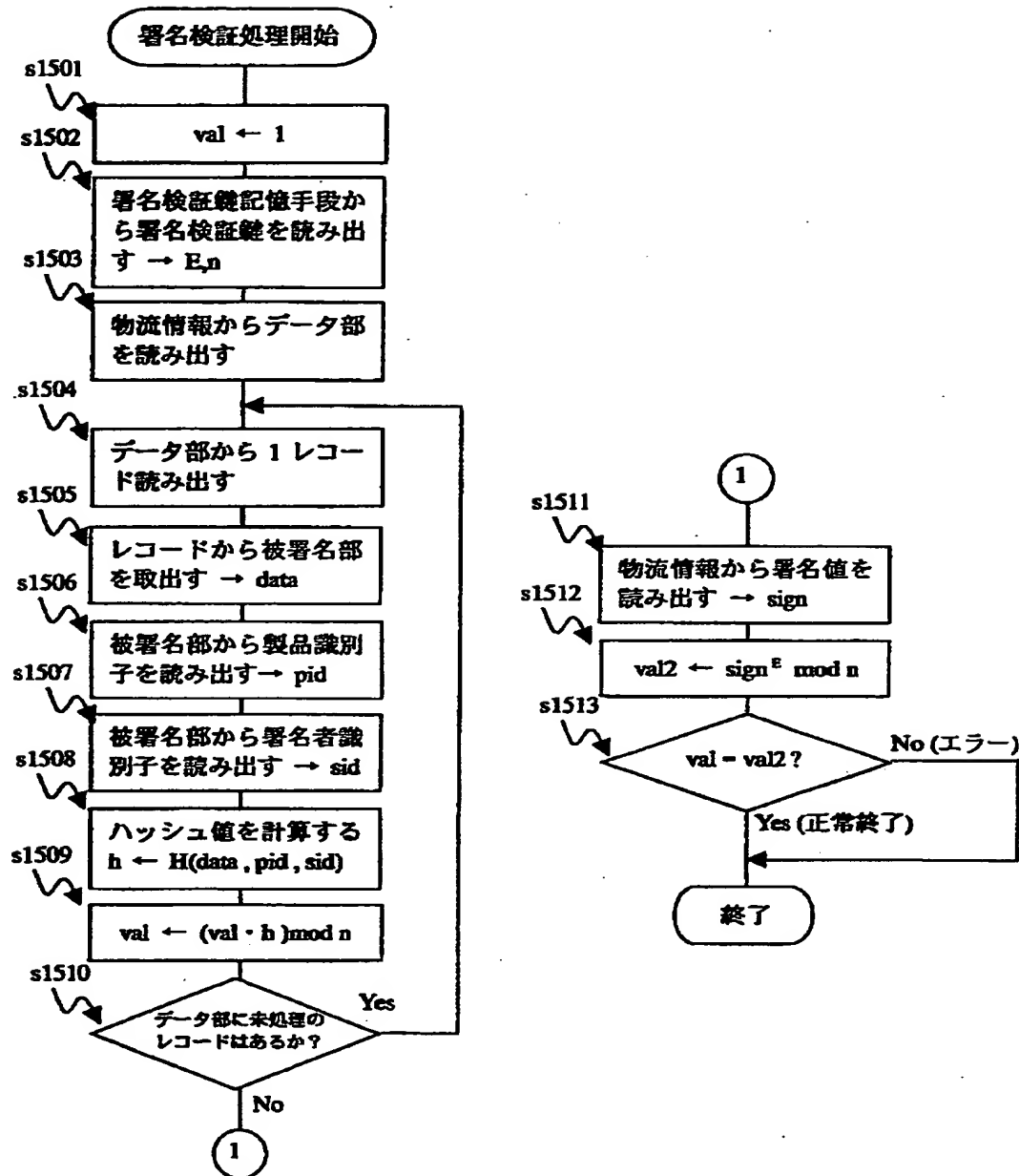


署名鍵情報取得時の処理

【図 14】

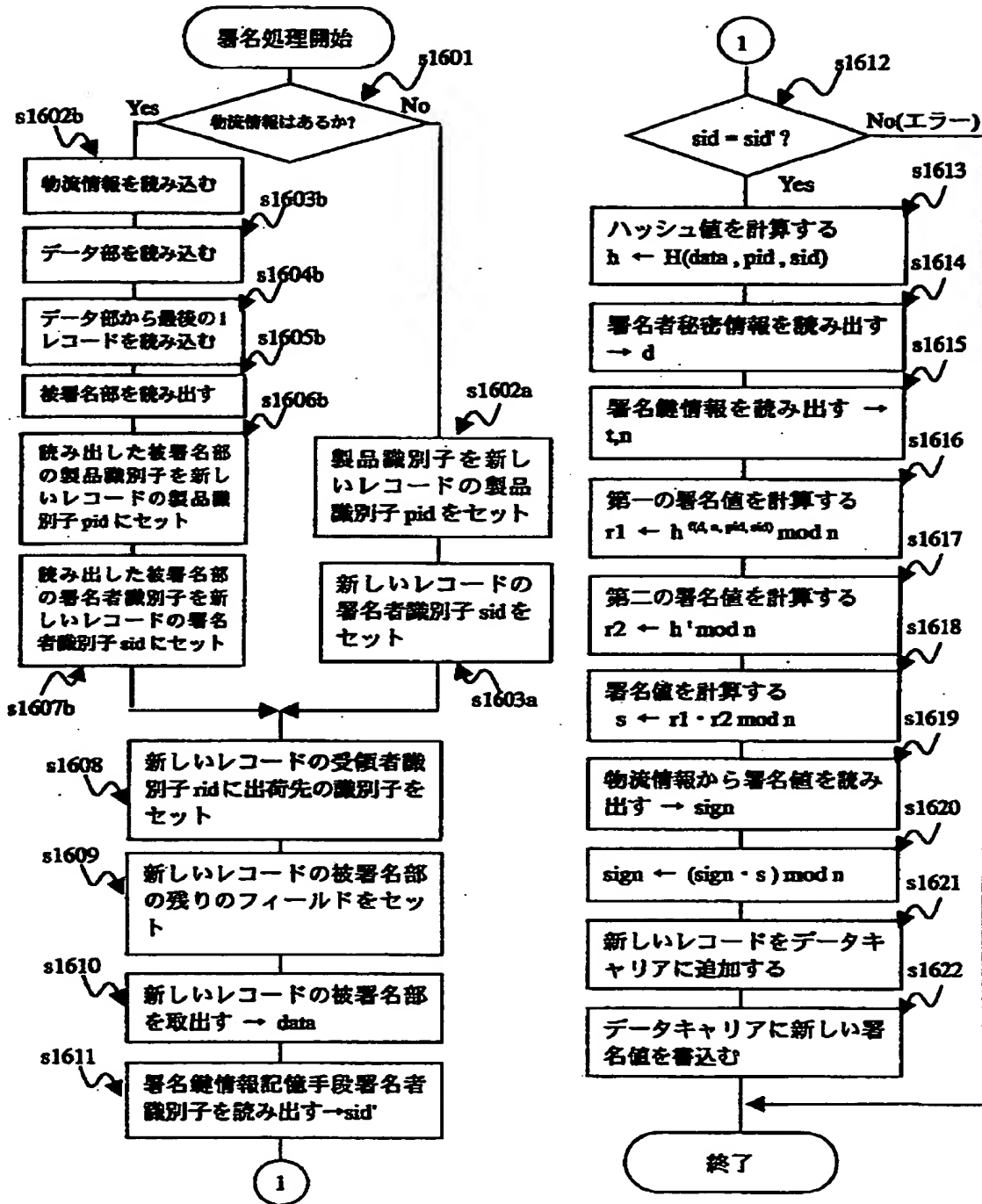


【図 15】



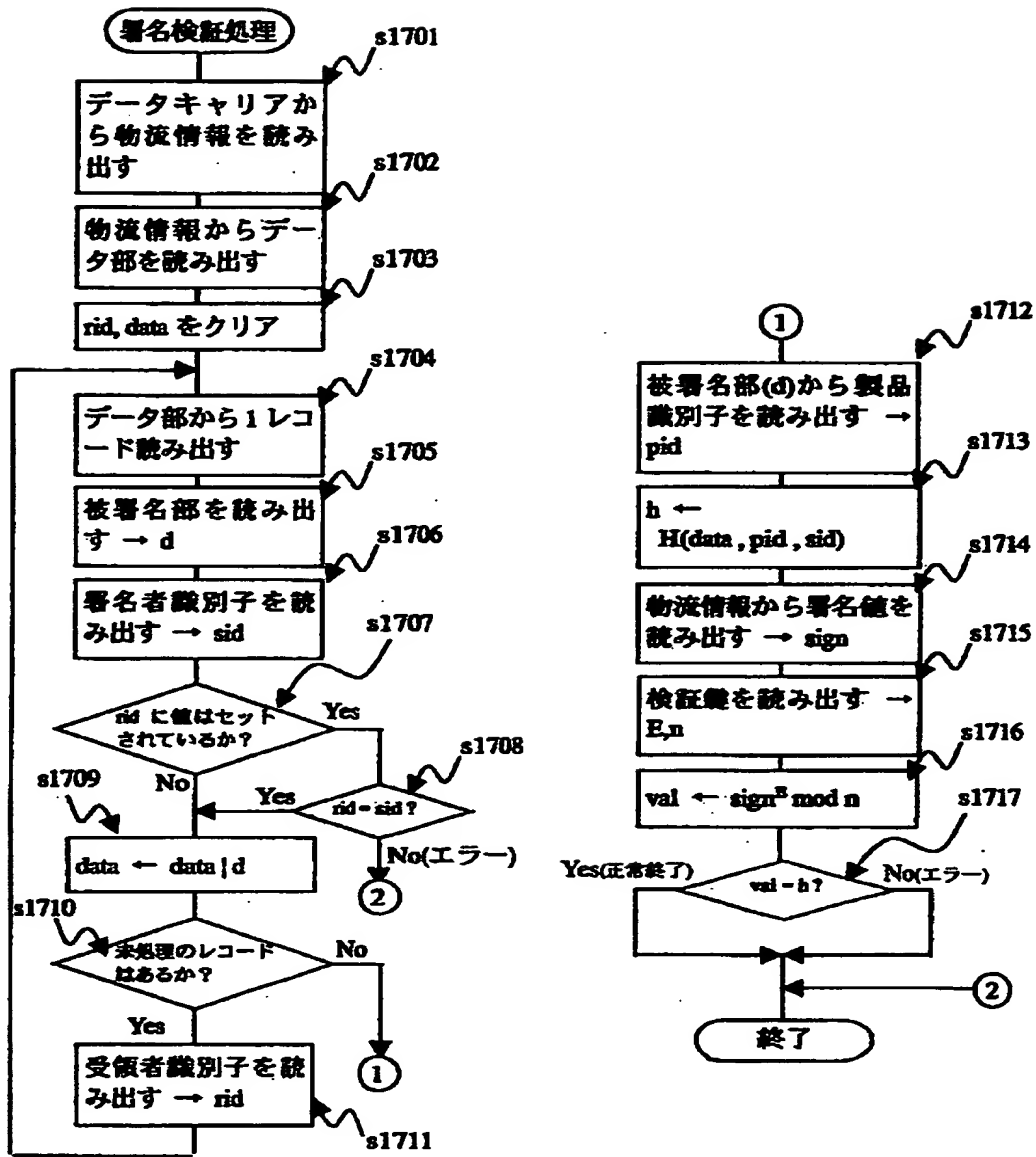
データキャリアに署名値を一つしか持たない場合の署名検証処理

【図 16】



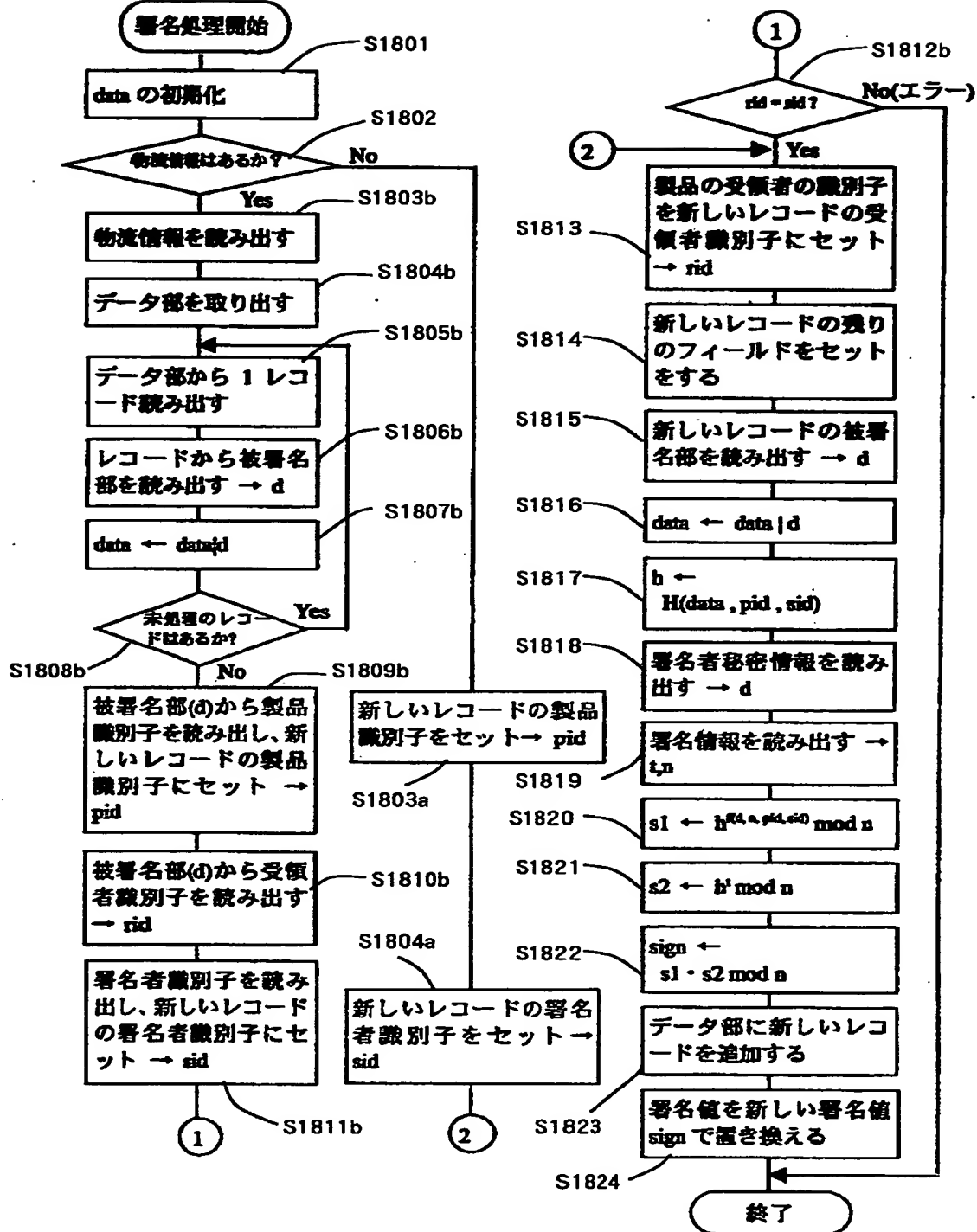
データキャリアに署名値を一つしか持たない場合の署名処理

【図 17】



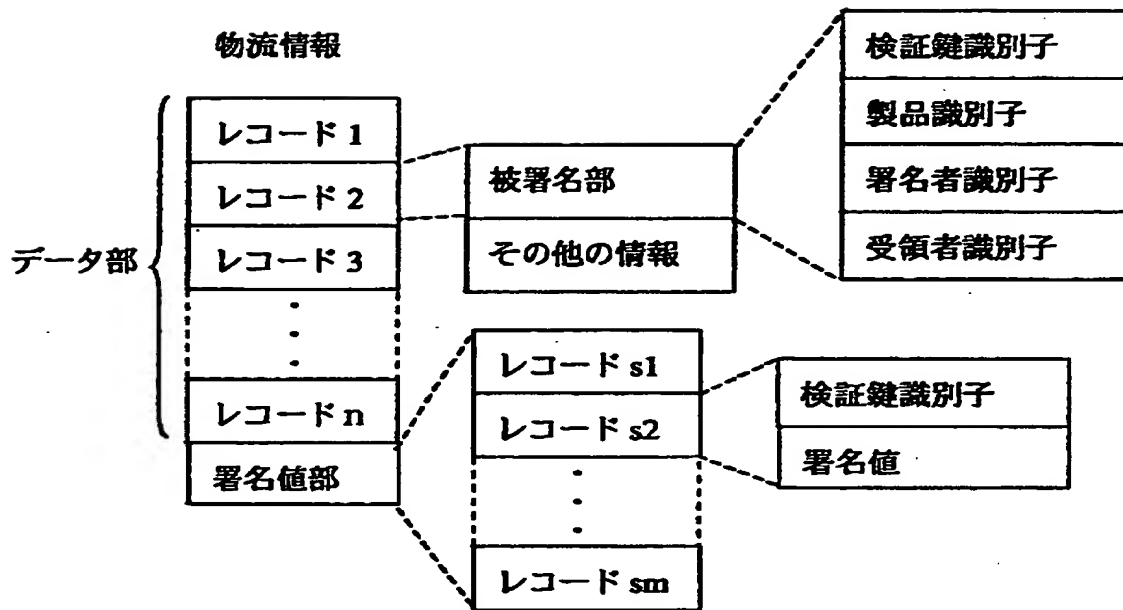
全体に署名をつける場合の検証処理（検証鍵は1つ）

【図 18】



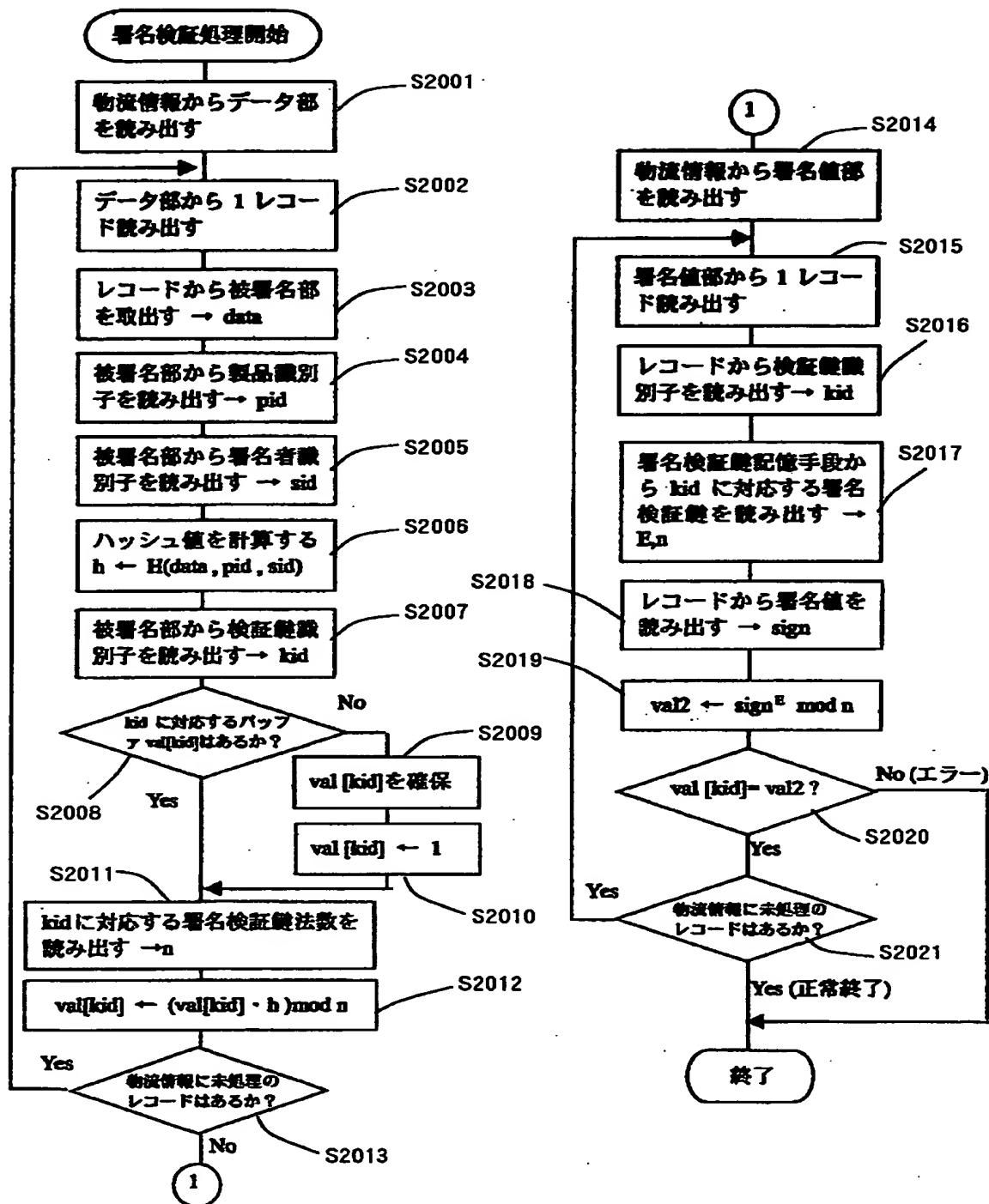
全体に署名をつける場合の署名処理 (検証鍵は1つ)

【図 19】



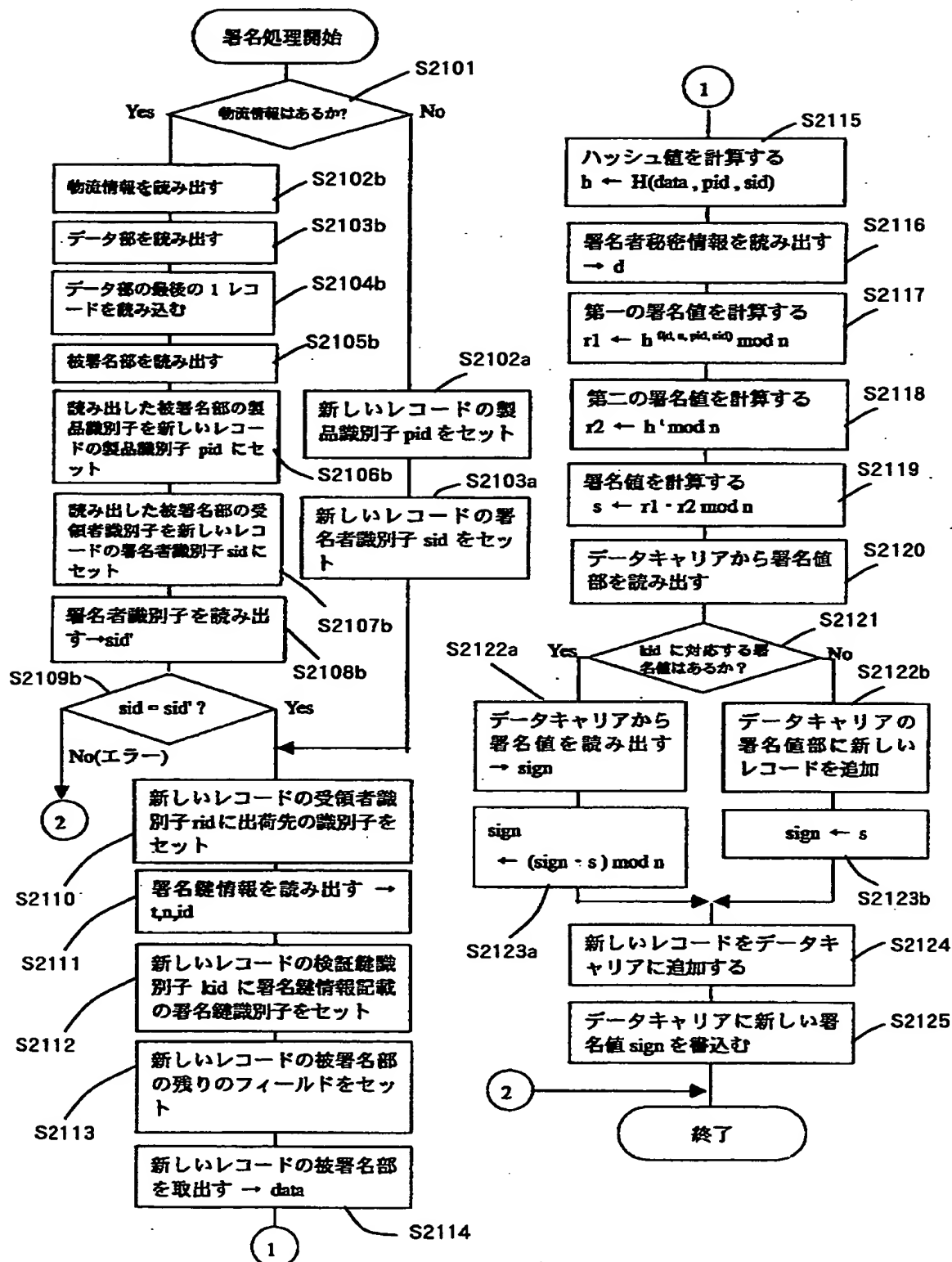
データキャリアに記憶される物流情報

【図 20】



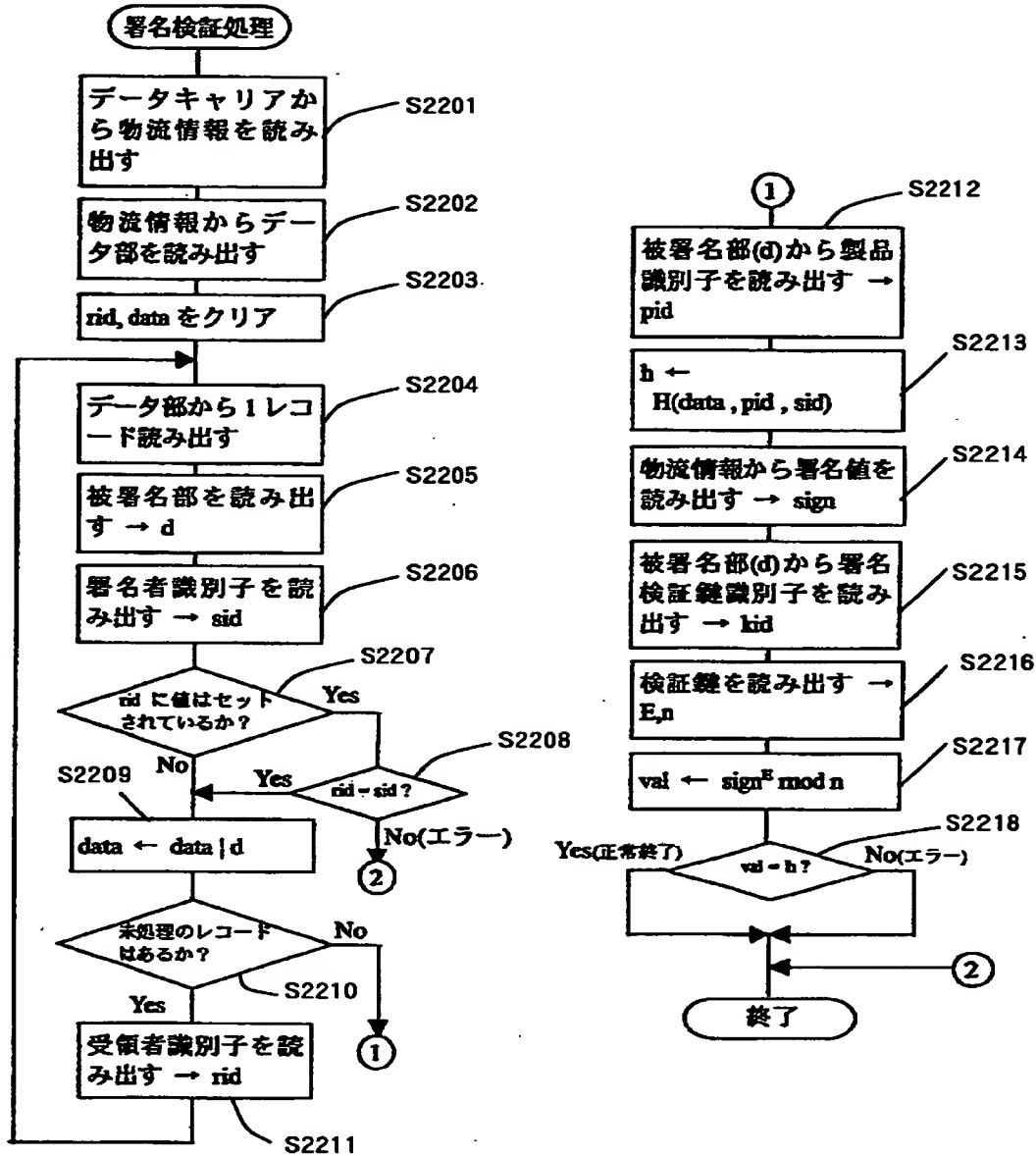
データキャリアに署名値を検証鍵毎に持つ場合の署名検証処理

【図 21】



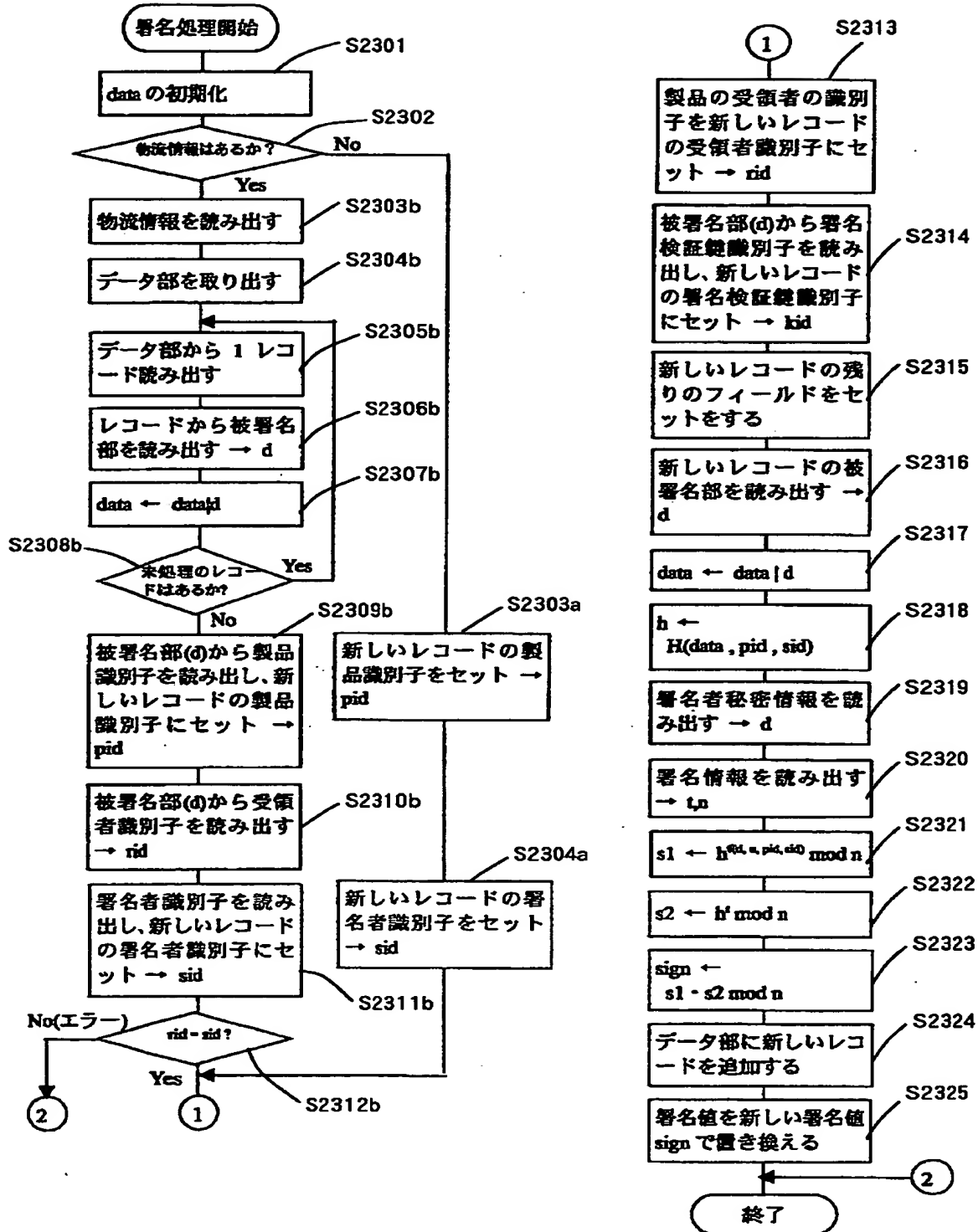
データキャリアに署名値を検証鍵毎に持つ場合の署名処理

【図 22】



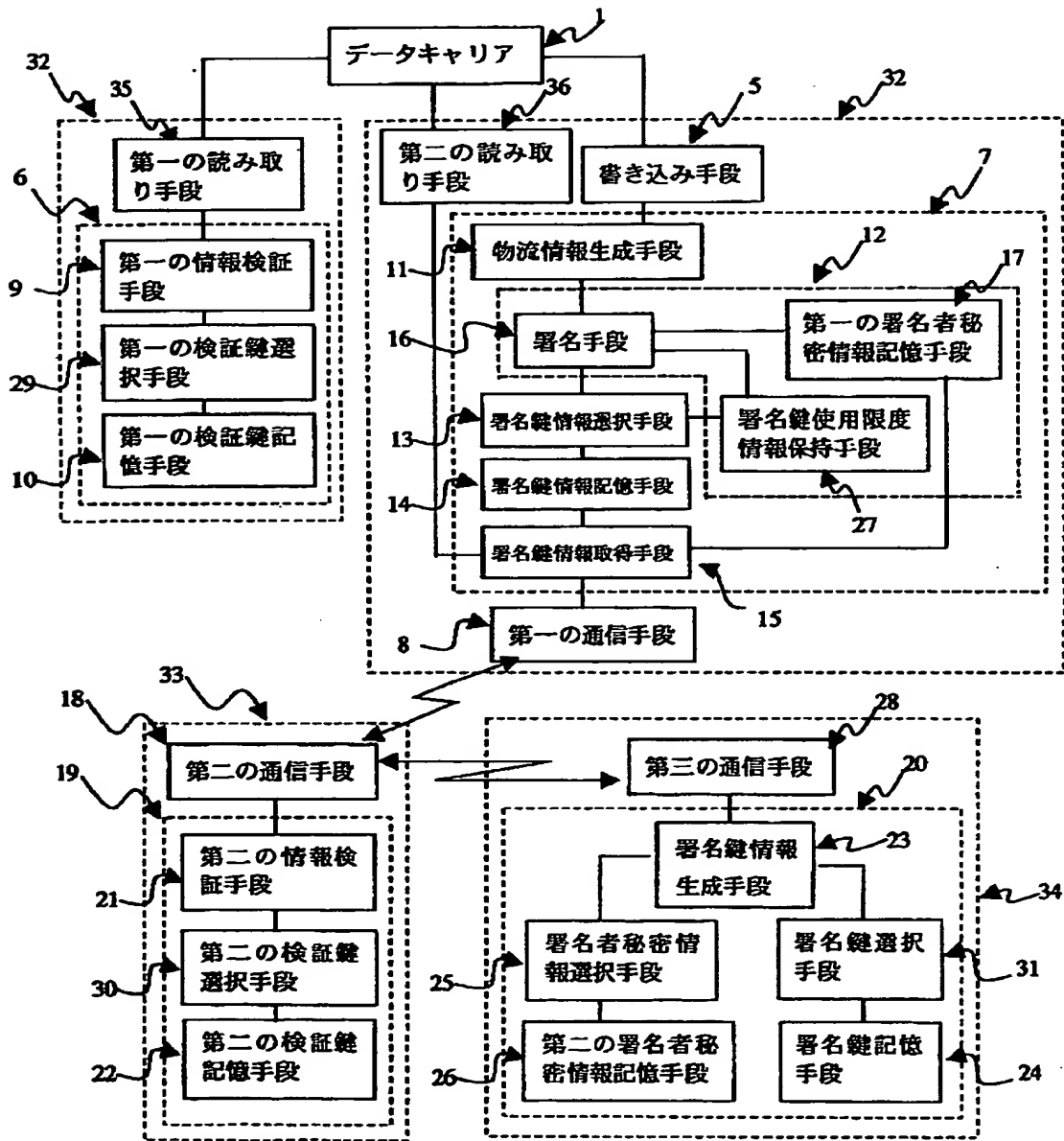
全体に署名をつける場合の検証処理（検証鍵は複数）

【図 23】



全体に署名をつける場合の署名処理 (検証鍵は複数)

【図 24】



変形例の構成

【書類名】 要約書

【要約】

【課題】 流通業者が、認証局から署名者ごとの証明書を取得する必要なしに、署名の検証を行えるようにする。

【解決手段】 署名手段 16 は、物流情報に対してハッシュ値を計算し、変数 h にセットする（ステップ S412）。第一の署名者秘密情報記憶手段 17 から署名者秘密情報を取り出し変数 d にセットする（ステップ S413）。署名鍵情報選択手段 13 は、署名鍵情報記憶手段 15 から製品識別子 pid に対応する署名鍵情報を取り出し、変数 t 、 n にセットする（ステップ S414）。署名手段 16 は、署名者秘密情報 d を用い変数 h に対して第一の署名値を計算し、変数 $r1$ にセットする（ステップ S415）。物流情報生成手段 11 は、署名鍵情報 t を用いて変数 h に対して第二の署名値を次式で計算する（ステップ S416）。計算結果 $r1$ と $r2$ を使って、最終的な変数 h に対する署名値を計算する。

【選択図】 図 4

出 願 人 履 歴 情 報

識別番号 [000005496]

1. 変更年月日 1996年 5月29日

[変更理由] 住所変更

住 所 東京都港区赤坂二丁目17番22号

氏 名 富士ゼロックス株式会社